A Biologically Plausible Mechanism for Phonology Acquisition in Human Infants

Max Strange

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

2019

Committee:

Michael Stiber, Chair

Yuval Marton

Dong Si

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

Abstract

A Biologically Plausible Mechanism for Phonology Acquisition
in Human Infants

Max Strange

Chair of the Supervisory Committee:
Professor Michael Stiber
Computing and Software Systems

Infants go through a serial developmental process in language acquisition, which is observable as specific linguistic milestones: by around six weeks of age, infants coo; by about six months they begin to babble, entering the reduplicated babbling stage at about eight months and non-reduplicated babbling at 11 months, with the first word not long after that. There are several theories of how infants do this, but few of them are end-to-end testable, often either falling short of being end-to-end, or else being too vague in certain details. This thesis presents a new theory of phonology acquisition, called the *Evolving Signal Chain* (ESC) theory of speech acquisition, which is extensible to lexical acquisition and which is end-to-end testable by means of computer simulation. Such a simulation is described and results from key portions of it are given. Key results include an analysis of deep convolutional autoencoders as a plausible means of learning an unsupervised encoding of raw speech and the use of that encoding in learning to reproduce natural speech. Results show that continuous natural speech can be encoded in at least two time scales and that these encodings can be used to recreate the beginnings of marginal babbling. Original contributions include the use of deep learning techniques in a computer model of primary language acquisition, and the use of an autoencoder instead of a self-organizing map, which allows for learning an optimal encoding space without specifying many *a priori* features.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACTIVATION FUNCTION: A (typically nonlinear) function used in neural networks to adjust input values into output values.

AUTOENCODER: A type of neural network that seeks to learn an identity function. Typically the first half of the network learns to encode the dataset and the second half learns to decode it. This is usually used to learn a new representation ("embedding") of the dataset (as well as new datapoints not seen in the dataset).

ALLOPHONE: See "phone".

BACKPROPAGATION: The typical training method used with neural networks. Neural networks are trained by computing the error value over a batch of training data by means of a loss function, then using backpropagation to adjust the weights of the network based on the error term. When used in combination with an optimization method, such as stochastic gradient descent, the error term may be minimized, and thus the network is trained.

CONVOLUTIONAL NEURAL NETWORK: A type of neural network that is used to detect patterns that are location-invariant.

COOING: The first non-fussing/non-crying vocalizations made by infants.

CORRESPONDENCE PROBLEM: The problem of infants having a different vocal anatomy from their caregivers, thus making it difficult to determine if they have made the "same" sound as one produced by their caregivers.

DISTINCTIVE FEATURE: A physical characteristic of a speech sound that distinguishes it from others, such as tongue placement.

GROUND-TRUTH: A dataset of labels that we can trust. Used to train supervised learning algorithms.

HEAD TURN PROCEDURE: A developmental psychology experiment paradigm for determining whether or not infants are capable of distinguishing between two stimuli (usually visual or audio).

HYPERPARAMETER: A decision that must be made at the outset when building a machine learning algorithm. For example, the number of layers and the size of each layer in a neural network are both hyperparameters. Hyperparameters are things that affect the behavior of a model, but which are not trained with the model — instead, they are hand-picked by the researcher.

IPA: The International Phonetic Alphabet. A writing system used to unambiguously describe pronunciation.

LOSS FUNCTION: A function that takes an output from a machine learning model and a desired output and returns some measure of error between the values.

LSTM: Long Short Term Memory. A type of recurrent neural network cell. This is useful for avoiding vanishing gradients, which may occur with simple RNN implementations, and for remembering inputs for longer than just the next step.

MARGINAL BABBLING: The beginning stages of babbling. Usually occurs starting around six months.

MORPHEME: The smallest unit of meaning, which may be sub-word, as in the case of "tele" in "television", or an entire word, such as in "cat".

MULTILAYER PERCEPTRON: A fully-connected, feed-forward neural network.

NATIVE LANGUAGE MAGNET THEORY (NLM): A theory of phonology acquisition in which clusters form in a perception embedding space in infants. The child learns to map these clusters to phonemes as heard in their native language, and these clusters serve as magnets for language sounds heard in the future.

NATURAL GROUP: A group of phonemes forms a natural group if they differ from one another in only one distinctive feature.

NONCE WORD: A nonsensical word that is well-formed phonologically, such as "flarble".

NONREDUPLICATED (VARIEGATED) BABBLING: The final form of babbling before a baby learns words. It is composed of complex, speech-like sounds.

PHONE: The physical (acoustic) embodiment of a phoneme. If this sound never occurs in the same environment as another sound (such as before a nasal consonant, or after a vowel), then the two sounds may be *allophones*.

PHONEME: A mental representation of a meaningful sound.

PHONOLOGY: A subfield of linguistics which is devoted to studying the sound systems of languages; phonology may also mean the sound system of a particular language.

PROSODY: The non-phonemic portion of speech which carries information. E.g. the rising intonation at the end of a question.

RECURRENT NEURAL NETWORK: A type of neural network used to detect patterns that occur over the course of sequences of input data. The network accomplishes this by having its outputs for time t fed back into it as inputs (along with its normal inputs) at time $t + 1$.

REDUPLICATED (CANONICAL) BABBLING: The form of babbling that is composed of repeated syllables. This is the first linguistic milestone that is not shared by deaf babies and hearing babies.

VARIATIONAL AUTOENCODER: A type of neural network used to generate new data points or cluster data samples.

ACKNOWLEDGMENTS

I want to express my most sincere appreciation for my thesis adviser Dr. Michael Stiber, who has helped reign in all the crazy ideas, my son, who was my unwitting accomplice in this endeavor, and my wife, whose patience clearly knows no limits.

I would also like to thank Nate True and Corey Wharton for taking time out of their days to mentor me and help my budding machine learning understanding along.

DEDICATION

To my son, Oliver, without whom I would have no reason to pursue this, and to my wife, Michelle, without whom I could not have done it at all.

Chapter 1

BACKGROUND: LINGUISTICS

Linguistics is the study of language, specifically with an eye towards understanding the underlying patterns and rules that dictate human communication. There are several subfields of linguistics, most of which represent efforts to understand a specific component of language. We will be mostly concerned with phonology and psycho/neurolinguistics in this thesis, but I give here a brief overview of the other major subfields of linguistics for context. Phonology, the study of the sounds of language, and psycho/neurolinguistics, the study of the mental/physical representation of language, are discussed in their own sections shortly.

Morphology is the study of *morphemes*, which are the smallest meaningful unit of speech. Every word is composed of one or more morphemes, which work together to create that word's meaning. For example, the word "television" is composed of "tele", meaning "remote" or "distant", and "vision". Contrast this with "telephone", which is composed of "tele" and "phone", where "phone" refers to sound. Another example is "hats", which is composed of two morphemes "hat" and "-s", which is the plural marker. It is not always as straightforward to determine which morphemes a word is composed of, such as in the case of "men", which is clearly derived from "man" and a plural marker, yet there is no "-s". Some words are composed of only one morpheme, such as "cat". We will not be concerned with morphology in this thesis.

Syntax is the study of a language's structure above the level of words. That is, syntax is the study of the rules that are allowed in the forming of sentences. There are many linguistic rules that humans internalize, but which they cannot necessarily articulate, such as why "The boy looked at the man with a telescope" is valid, even though it is ambiguous, but "The with a boy telescope man looked at the" is invalid. The field of syntax seeks to discover

these rules and, to some extent, understand why or if they are necessary, as well as to what extent they are universal.

Semantics is the study of the meaning of words and utterances. It relates the form of words to their meaning. Pragmatics, meanwhile, is the study of the high-level organization of speech or writing (across sentences), as well as the study of the meaning of entire utterances or documents. For example, it is possible to say "it sure is cold in here", and have the intention of the utterance be that someone close the window.

Beyond theoretical linguistics, there are also more applied branches of linguistics, such as computational linguistics, which is the study of computer science as it relates to natural language, historical linguistics, which is the study of how languages have been used and changed over time, sociolinguistics, which is the study of how languages are different in different societies, and many others.

With this brief survey of linguistics out of the way, we now turn to phonology and psycho/neurolinguistics. Note that language acquisition (the learning of language) will be covered in the next chapter.

1.1 Phonology

Phonology is the study of the sound systems of languages. In contrast to *phonetics*, which is the study of the physical production, transmission, and sensation of the sounds of a language, phonology is concerned with the underlying psychological constructs (called *phonemes*) which give rise to the specific sounds (or *phones*) of a language. That is, phonology is the study of the sounds of a language as they occur in a human brain both before an utterance is spoken and after that utterance is heard. In this way, the phonology of a particular language is the mental representation of the sounds of that language.

1.1.1 Phonemes and Phones

Figure 1.1 shows the difference between phonemes and phones. This difference may be summed up like this: what we *hear* is phones, but what we *think* we hear is phonemes. And

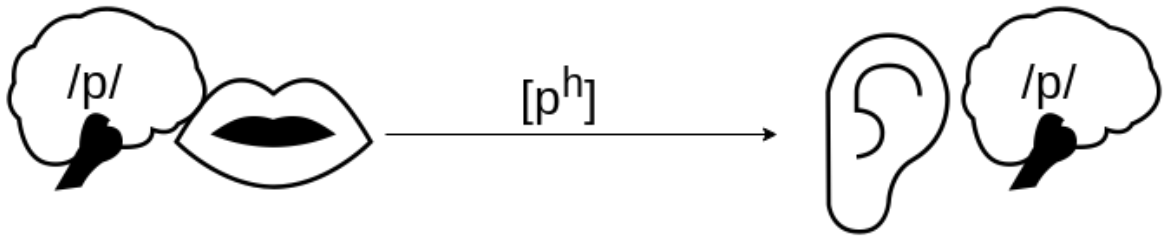

Figure 1.1: According to traditional linguistic theory, phonemes are what occur in the brain, while phones are what are actually said and heard. Phonemes are traditionally shown in slashes, while phones are shown in square brackets. In this case, we have a phoneme, "p", which is uttered (and heard) as "p" followed by a little puff of air.

vice versa: we *say* phones, but we *think* we say phonemes. The difference is important. To borrow from information theory for a moment, (according to conventional linguistic theory), phonemes are the symbols that make up the code that is a particular language. When a speaker utters something, they encode phonemes into sound, mixing in things like aspiration and nasalization to create phones. This is done unintentionally, but not randomly. Almost all speakers of a particular dialect of a language will produce the same sequence of phones when trying to say the same thing, even though almost none of them are aware they are doing things like vowel nasalization or plosive aspiration. The reverse process, decoding, is done by a listener - they hear the phones but, according to linguistic theory, they decode the phones into phonemes. Whether this is strictly true in the brain of speakers and listeners or whether phonemes are a purely theoretical construction is a matter of some debate (which I will weigh in on later in this work).

1.1.2 Distinctive Features and Allophones

Underlying phonemes become different phones in different contexts. For example in English, the /k/ phoneme becomes aspirated before a vowel unless it is preceeded by an /s/. The different phones for a single phoneme are called that phoneme's *allophones*. In this example, [k^h] and [k] are both allophones of the phoneme /k/ (traditionally, linguists transcribe

phonemes using slashes and phones using square brackets, to distinguish between them). Allophones are heard by native speakers as the same sound, even though they are demonstrably not the same. The reason for this is again that a native speaker maps both sounds to the same underlying phoneme.

Importantly, what are allophones (and therefore heard as the same sound) in one language may be completely different phonemes in another. For example, while [k] and [k^h] are allophones in English, and therefore the average English speaker will have a hard time telling them apart, they are actually separate phonemes — i.e., they are heard as completely different sounds — in Hindi. The sounds [k] and [k^h] are as different to a speaker of Hindi as the sounds [k] and [g] are to a speaker of English. This means that an infant must be able to learn a wide variety of phonemes from his or her linguistic environment. One possibility is that infants are tuned to what are called *distinctive features*. Distinctive features are the physical differences between speech sounds that make them map to different underlying phonemes, such as location of the tongue, whether the vocal cords vibrate, etc. Groups of phonemes in a language form a *natural group* if they differ in only one feature. For example, the phonemes /p/, /t/, and /k/ in English form a natural group since they only differ in the place of articulation: /p/ is pronounced with the lips, /t/ is pronounced with the front of the tongue, and /k/ is pronounced with the back of the tongue - all other distinctive features are identical amongst them.

An interesting question is how might a linguist go about drawing up a list of all the phonemes in the English language? The answer is *minimal pairs*. If two words (with different meanings) can be found in a language which differ in exactly one phone, then that difference must make a pair of phonemes. For example, [slæp] ("slap") and [slæb] "slab" differ only in the [p] and [b] sound, yet they mean two different things. Therefore, [p] and [b] must constitute phonemes (/p/ and /b/).

Table 1.1: Vowels in the IPA with American English pronunciation examples.

[a]	"ah" as in "lot"
[æ]	"aa" as in "bat"
[ɛ]	"eh" as in "bet"
[e]	"ey" as in "late"
[ɪ]	"ih" as in "bit"
[i]	"ee" as in "eat"
[ə]	unstressed "uh" in "banana"
[ʌ]	"uh" as in "under"
[o]	"oh" as in "over"
[ʊ]	"uh" as in "wood"
[u]	"oo" as in "boot"

1.1.3 The International Phonetic Alphabet

The *IPA* (*International Phonetic Alphabet*) is a system of writing which attempts to unambiguously describe pronunciation. Figure 1.2 describes the IPA [2].

The most important parts of this chart are the top portion (the consonants) and the vowels, which are on the right. The sounds that are most likely to occur in this thesis are presented in Table 1.1 and Table 1.2 with an attempt at a layman's explanation of what is present in the IPA chart, using an American accent.

Here is an example of phonetic transcription using the IPA:

[ðɪsɪzənaip^{h}iaitɹæ̃nskɹɪpʃn̩] "This is an IPA transcription"

In contrast to phonetic transcription, there is also phonemic transcription. In the above (phonetic) example, every sound that the speaker makes is transcribed and characterized, including the aspiration of the "p" (the little puff of air that an English speaker unintentionally places after most [p], [t], and [k] sounds) and the nasalization of the "a" in "transcription".

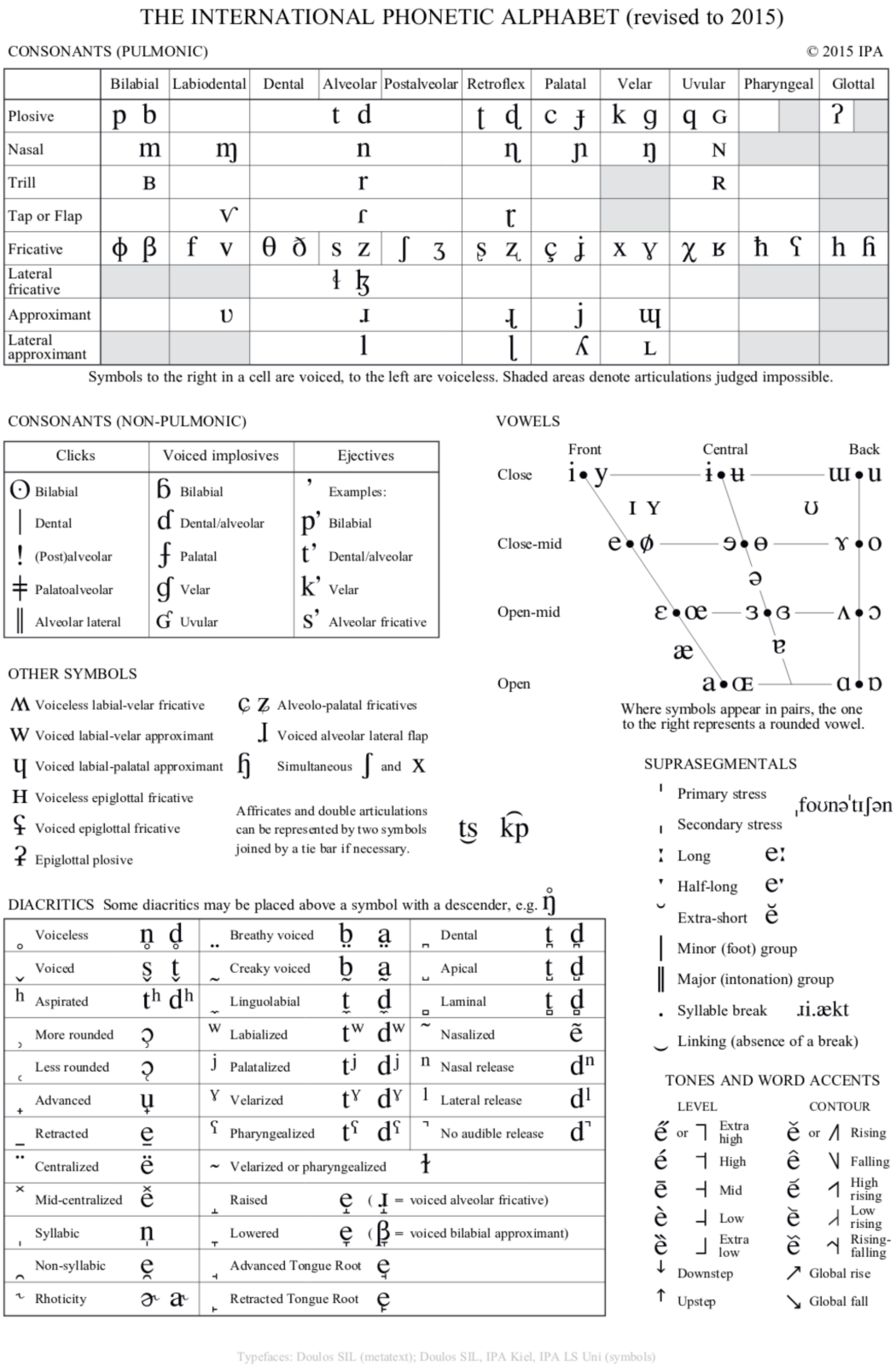

Figure 1.2: Table of IPA symbols [2].

Table 1.2: Consonants in the IPA with American English pronunciation examples.

[p]	"p" as in "past"	**[θ]**	"th" as in "thought"
[b]	"b" as in "be"	**[ð]**	"th" as in "they"
[t]	"t" as in "to"	**[s]**	"s" as in "so"
[d]	"d" as in "do"	**[z]**	"z" as in "zoo"
[k]	"k" as in "kid"	**[ʃ]**	"sh" as in "shoe"
[g]	"g" as in "go"	**[ʒ]**	"zh" as in "treasure"
[m]	"m" as in "me"	**[h]**	"h" as in "hot"
[n]	"n" as in "no"	**[ɹ]**	"r" as in "round"
[ŋ]	"ng" as in "long"	**[l]**	"l" as in "long"
[f]	"f" as in "food"	**[j]**	"y" as in "year"
[v]	"v" as in "very"	**[w]**	"w" as in "with"

But there is also phonemic transcription, which attempts to record the phonemes which are meant by the speaker, rather than the phones that the speaker actually uttered. In phonemic transcription, the above listing would be:

/ðɪsɪzənaipiaitɹænskɹɪpʃən/

1.2 Neurolinguistics and Psycholinguistics

The last fields of linguistics I want to touch on in this chapter are the fields of *psycholinguistics* and *neurolinguistics*. Psycholinguistics seeks to provide theoretical cognitive models of how language is represented, understood, and constructed with respect to the the human mind. Psycholinguists study how language might work from a cognitive psychology point of view. They produce high-level models of speech production, perception, lexical access, and representation. Psycholinguists may also theorize how infants acquire language (such as is studied here in this thesis). Psycholinguists use a range of methods, from introspection to

experimental studies, including eye-tracking and reaction times under different psychological testing paradigms.

Neurolinguistics on the other hand, is the study of how language is represented in the brain. It is similar to psycholinguistics, but it is more grounded in biology and biological evidence, such as that obtained by observations of patients with brain damage or through brain imaging studies. This thesis is relevant to neurolinguistics in that it attempts to provide a biologically plausible explanation of how the phonology of a language is acquired by an infant. A typical question that neurolinguistics seeks to answer (and which has relevence to this work) is where and how exactly the brain represents the sounds of a language. It turns out that the first part of this question is easily answered - or at least, it is easy to deduce areas of the brain that are necessary to the processing of language (even if this does not prove that sounds or words are *stored* there), but that there is significant disagreement about the answer to the second part, with some studies providing evidence that sounds are represented in the brain at the level of phoneme [57] and others arguing that they are represented at the level of distinctive feature [51]. These of course, are not the only possibilities, but they are the most prominant opinions amongst linguists.

I refer the reader to [48] for a good discussion of psycholinguistic theory and findings from a prominent figure in the field, and [18] for a broad survey of the field of neurolinguistics.

Chapter 2

BACKGROUND: DEVELOPMENTAL PSYCHOLOGY

Developmental psychology is a subfield of psychology devoted to studying how humans develop mentally from infants to adults and then into old age. Developmental psychology is a fascinating field of study that has brought us many ideas that are present even in the mainstream, such as the ideas of nature vs. nurture or the work of Jean Piaget. While this field contains myriad interesting studies and findings, we will mainly concern ourselves with its work on language acquisition in children.

2.1 Methodology

How does a human infant learn language? The fact is that we don't know. Language is extraordinarily complicated, and ethical concerns (rightly) prevent the types of studies that would be required to arrive at an answer to this question without a great deal of conjecture. What we do know has come from a few different study paradigms. This section outlines how some of these paradigms work.

2.1.1 Infant Head Turn Procedure

For the studies that are of most interest for this thesis, the *conditioned infant head turn procedure* is the paradigm that is used most often. This type of study allows the psychologist to determine whether infants are able to tell the difference between two stimuli (usually audio or visual). First introduced by Dix and Hallpike in 1947, it has been modified and further refined into its modern form [69].

The head turn procedure works as follows: an infant is seated on a caregiver's lap, looking straight ahead at an experimental assistant who is doing visually salient things (playing with

a toy, smiling, waving, etc.) in order to keep the child's attention. On one side of the infant is a screen that is typically blank. The room is equipped with a speaker. Over the speaker, the child will hear different sounds. The child is trained to look at the blank screen when they are able to discern a difference between two things played over the speaker. When there is a difference between sounds played back-to-back on the speaker, and the child looks at the screen, the screen will play something that the child finds enjoyable and the assistant will praise the child. If the child looks at the screen when there is no difference however, their behavior is ignored. Therefore the child is rewarded whenever they succeed at distinguishing between sounds played over the speaker. To prevent contamination of the child's behavior by the caretaker or assistant, both are wearing headphones that play music loud enough to block out the speaker. In this way, the only person in the room who can tell when it is time to look at the screen is the infant, and therefore the infant cannot take any social cues. To prevent contamination of the sounds played over the speaker, the room is insulated so that the only sounds that can be heard are those that come over the speaker. This paradigm draws heavily from behaviorist principles to shape the child into a conditioned response based on stimuli, and just like with lab rats in Skinner boxes, the infant goes through a training phase before they graduate to the test procedure.

There are many variations of the conditioned infant head turn procedure, including ones that allow for newborns to be tested, but this outline will suffice for our purposes. It is important to note though, that while this procedure allows us to tell whether or not an infant can tell the difference between two auditory stimuli, human infants are difficult to get in large numbers for psychological studies, and therefore the statistical power of these tests is often lacking. Thus care must be taken when interpreting the results of such a study, and only after findings are reproduced should they be taken with less than a healthy dose of salt.

2.1.2 Brain Imaging

Another paradigm of research in the psychology of language is brain imaging. Two types of brain imaging technologies in particular are used: positron emission tomography (PET),

and functional magnetic resonance imaging (fMRI). PET and fMRI both operate under the assumption that an increased blood-oxygen supply to a particular region of the brain indicates that that region of the brain is under heavy use. A typical study using brain imaging will take a baseline image of a study participant's brain, then the participant will be asked to take part in a task, such as a sentence processing task. The participant's brain is again imaged, and the baseline is subtracted from this new image. By applying this method to several participants, researchers can glean information about the locations in the brain that are utilized most heavily during particular tasks.

Although brain imaging studies offer great insight into adult brains, they have found limited use in developmental psychological or linguistic research with infants [6]. The reasons for this mostly stem from infants being unable to follow directions or hold still for the extended periods of time required to obtain good results from current neuroimaging technologies.

2.1.3 Computational Models

The final paradigm that we will discuss here is that of computational modeling. Computational modeling is the production of a computer simulation to model a physical or biological process. It is often used in fields where experimentation would be exceedingly difficult, such as with weather patterns, climate change, or evolution, and in fields where experimentation can be ethically dubious, such as developmental psychology. Language development in particular lends itself to computer simulation since simulations of language acquisition can be checked against the many well-documented and observable linguistic milestones that children display as they learn their first language(s). That is to say that the process of language acquisition follows a fairly deterministic time course, with many observable changes along the way. Although we do not know the internal process that leads to things like laughter at sixteen weeks of age and babbling at six months, we can program a computer model, feed that model the same data that a human would get, and see if it displays the same behavior as we know humans do.

Unfortunately, even if we program a model, feed it data, and find that it displays the

Table 2.1: Linguistic milestones and their typical onset times. Onset times vary substantially for milestones occuring after about six months.

Cooing	6 weeks
Laughter	16 weeks
Marginal Babbling	6 months
Reduplicated Babbling	8 months
Nonreduplicated Babbling	11 months
First Word	12 months

same behavior as humans at around the same (simulated) times, this merely tells us that the model *may* be the way humans learn language. If the model does *not* work, this provides strong evidence that the model is *not* how humans learn language, but if the model *does* work, it only provides circumstantial evidence that the model is correct. However, a successful computer model can suggest new experimental directions to try. If a computer model succeeds at modeling a particular language acquisition phenomenon, it provides developmental psychologists with a working theory that they may use to design new studies to test the model's assumptions. If researchers are unable to prove that the computational model is not correct despite testable claims that it makes, it provides strong evidence that this is the way it works in humans.

Lastly, computational models have the benefit of adding not only to the developmental psychology literature, but also to that of computer science. A successful computational model may be used for engineering applications or for research into areas like machine learning and artificial intelligence.

2.2 Linguistic Milestones

Table 2.1 outlines what children are capable of, and at what age, in terms of language acquisition [14]. Please note though that while each of these milestones occurs in healthy

children in this order, the actual dates are subject to a large degree of variance. It is not uncommon for children to have their first word at 18 months, for example.

The first vocalizations that a newborn infant makes (besides the in-born ability to fuss, cry, sneeze, burp, etc.) are called *coos*, and are just that - vocalizations. While essentially random and uncontrolled, coos do seem to be more common during social interaction [14, 45]. After cooing comes laughter, and then *marginal babbling*, which is the very beginning of babbling. Marginal babbling is composed mostly of vowels and consonants produced with the back of the tongue, such as [k] and [g]. After marginal babbling comes canonical or *reduplicated babbling*, which is composed of repeated (reduplicated) series of vowel-consonant pairs, like [babababababa] or [gʌgʌgʌgʌgʌ]. Babies will make these noises even when isolated, such as when they are in a crib (when they should be sleeping!). Interestingly, deaf children also babble. However, they do not engage in reduplicated babbling. Their vocalization timeline is indistinguishable from hearing children up to this point [43].

After reduplicated babbling comes true babbling, variegated, or *nonreduplicated babbling*, which is composed of non-repetitive speech-like syllables, like [badaguga], and even contains *prosody*, the non-phonemic information-carrying portion of speech, such as tone. Finally, sometime during the second year of a child's life, they will utter their first word. A child's first word is the first sound that they make in an effort to actually mean something, either pragmatically (like "yes", "no", or "up") or referentially (like "dog" or "mommy").

Chapter 3

BACKGROUND: COMPUTATIONAL MODELING

This thesis is a computational model of how infants may learn the phonology of their primary language(s). It draws heavily on certain classes of software algorithms which I describe below. These algorithms all fall under the larger umbrella of artificial intelligence, and more specifically, under the subfield of machine learning. Machine learning is traditionally thought of as being composed of three types of algorithms: supervised learning algorithms, unsupervised learning algorithms, and reinforcement learning algorithms.

All of these algorithms have in common that they use statistical means to accomplish what is impractical using more traditional if-else style logic. This leads to a few byproducts. First, because these algorithms are more statistical and heuristic than traditional ones, they may sometimes bear a resemblance to the decision-making processes of biological agents. For example, convolutional neural networks are actually inspired by how the mammalian vision system works. Often, this is merely a superficial similarity, but it is nonetheless an interesting observation in the context of using these methods for constructing computational models of biological systems. The second byproduct of these algorithms using statistical methods to solve problems is that they are often non-deterministic in the sense that the same algorithm may have different behavior in almost identical contexts, based merely on the inherent randomness of the underlying model. For example, supervised learning algorithms require a training phase, where instances of correct answers are revealed to the algorithm one at a time. The order in which these items are shown to the algorithm often matters, but not in ways that are easy to understand or reproduce. Lastly, machine learning algorithms are notoriously difficult to get working for more than trivial examples. They have a great many "knobs", called *hyperparameters*, which must be tuned expertly. But the tuning of these

hyperparameters is, even for expert practicioners, sometimes little more than guesswork.

3.1 Supervised Learning

Supervised learning algorithms are ones which work by producing a guess which is then checked against the right answer. The difference between the guess and the correct answer is used to create an error signal which is then fed back into the algorithm in order to make future guesses closer to the desired answers. Examples are neural networks, support vector machines, and decision trees. This thesis makes heavy use of neural networks. Throughout supervised learning literature, and especially when discussing neural networks, the following mathematical notation is used:

- $\mathbf{x}$ - An input vector
- $\mathbf{y}$ - A label vector; i.e., the correct answer for the input $\mathbf{x}$
- $\hat{\mathbf{y}}$ - An output vector; i.e., the guess the algorithm comes up with for input $\mathbf{x}$
- $\mathbf{X}$ - All input vectors in a dataset or batch of data
- $\mathbf{Y}$ - All labels for a dataset
- $\hat{\mathbf{Y}}$ - All guesses for a dataset

In general, the purpose of any supervised learning algorithm is to learn a function f of an input vector $\mathbf{x}$, parameterized by a vector of trainable parameters, $\boldsymbol{\theta}$, such that whenever the function is presented with a new sample of data, the output vector, $\hat{\mathbf{y}}$, is close to the correct answer, $\mathbf{y}$. That is:

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{3.1}$$

3.1.1 Neural Networks

Neural networks are a class of algorithms originally inspired by Hebbian learning to mimic biological neural networks. They make use of nonlinear equations to approximate a (potentially nonlinear) function. A neural network takes a vector of floating point values $\mathbf{x} \in \mathbb{R}^n$

and outputs a vector of floating point values $\hat{\mathbf{y}} \in \mathbb{R}^m$, where n and m are integers and may be the same value. The pseudocode for training a typical neural network is given in Algorithm 1.

```
Data: X, list of input vectors
Data: Y, list of desired output vectors in same order as X
Data: loss_function
Data: Nepochs, the number of complete cycles through the data
Result: A trained neural network
initialize all weights in network to small random numbers;
for epoch in Nepochs do
    for (x, y) in (X, Y) do
        ŷ = network.forward_pass(x);
        error = loss_function(ŷ, y);
        network.adjust_weights(error);
    end
end
```

Algorithm 1: High-level pseudocode for training a neural network.

Fundamentally, a neural network is a network of nodes (often called "neurons") arranged in layers. Vectors come in via the input layer, are operated on at each subsequent layer, and are then output at the output layer. Each node sums the values entering it and then applies a nonlinear *activation function* to the result before passing it on to whatever nodes it is, in turn, connected to. Each connection between nodes is assigned a weight, which is multiplied against the value going over the connection. These weights are initialized as small random values, but are tuned over the course of training via a process called *backpropagation*. As these weights change, so does the output vector for a given input vector, until the accuracy of the network has converged on 100% (ideally - though in reality, 100% accuracy would, for most problems, mean that the network had *overfit* the training set, a topic we will not be

discussing).

Algorithm 1 in words goes like this: for some number of times through the whole dataset, get the next input $\mathbf{x}$ and desired output $\mathbf{y}$ from the dataset. Use the input $\mathbf{x}$ on the network to calculate its current estimate $\hat{\mathbf{y}}$. This procedure is called the *forward pass*. Next, a *loss function* (also called a *cost function*) is used to compare the desired output $\mathbf{y}$ with the actual output $\hat{\mathbf{y}}$ in order to produce an error term. This error term is then used in backpropagation to determine how much error each node in the network contributed to the result. The error for each node is then used in an optimization algorithm (typically some derivative of *gradient descent*) to produce a change to the weights coming into each node. These last two steps are covered in the *adjust_weights* function in Algorithm 1.

A few things are worth noting about this algorithm. First, although this algorithm may work as written, there is a problem with it. Specifically, if we compute the error term from each individual data point in the entire dataset and do a backpropagation pass with it, it can potentially take a very long time. Not only this, but it also will make the network adjust its weights in an over-corrective fashion, like making large adjustments to the steering wheel of a car in response to every single visual stimulus. A better alternative would be to do a forward pass and apply the loss function (both of which are relatively computationally inexpensive) over the whole dataset, and then use some form of averaging over the resultant error. Then use that error in the backpropagation step. This would make for a much smoother descent down the gradient, but with very large datasets, this would also take far too long to be practical. Instead, deep learning practitioners typically employ a technique called *mini-batching* and compute an average error over some small number (relative to the size of the dataset) of datapoints, called a *mini batch* (or more often, simply a *batch*). This strikes a balance between following a smooth gradient (i.e., not over-correcting) and training the network in a reasonable amount of time, even on very large datasets (potentially numbering in the hundreds of millions of datapoints).

A second thing to expand on in regards to this algorithm is the *loss function*. The loss function takes an actual output $\hat{\mathbf{y}}$ and a desired output $\mathbf{y}$ and produces some measure of the

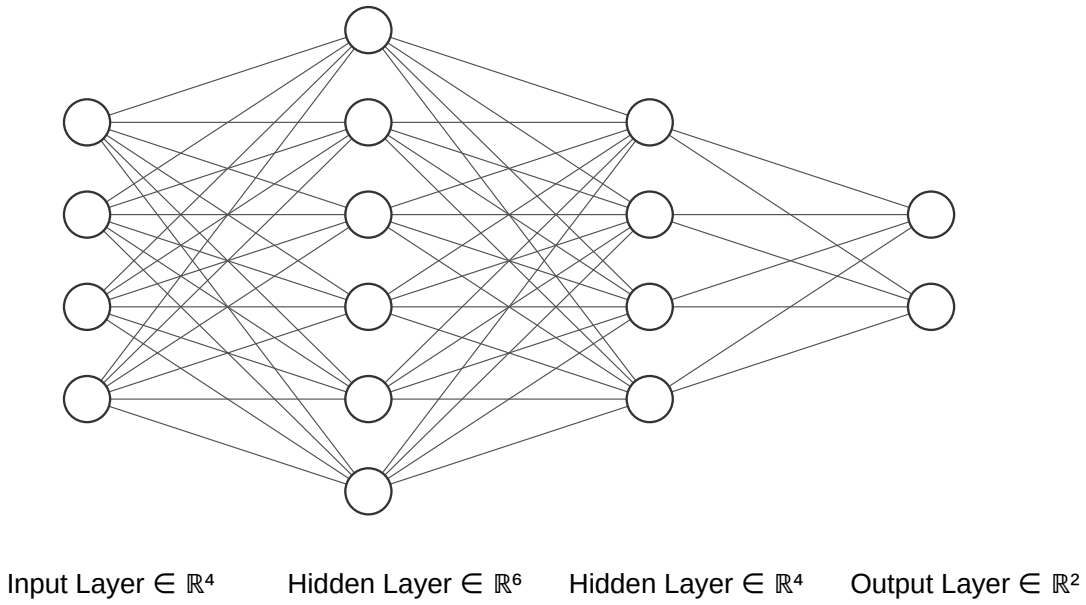

Figure 3.1: A Multilayer Perceptron (MLP) [27].

difference (or error) between the two. There are many loss functions in use today, but the choice is mostly up to what type of model is being trained (probabilistic, categorical, etc.) rather than anything else.

Figure 3.1 shows a 3-layer network (in counting layers, you typically do not include the input layer). This network has two *hidden layers*, the first of which contains six nodes, and the second of which contains four. The input to this network is a vector in $\mathbb{R}^4$, and the output is a vector in $\mathbb{R}^2$.

Because neural networks are simply nodes with connections between them, there are a great many different ways to lay them out. The most traditional type is a fully-connected, feed-forward neural network, or *multilayer perceptron (MLP)*. Fully-connected means that each neuron in layer i is connected to each neuron in layer j, and feed-forward means that all the connections are in the same direction so that an input signal flows through it to the output layer without any cycles. Below, I describe the different architectures of neural networks used in this thesis.

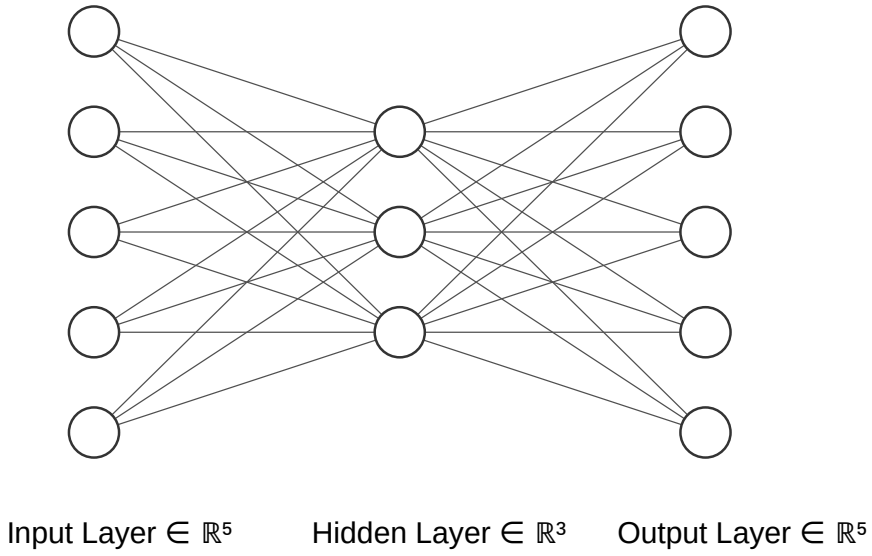

Figure 3.2: An typical autoencoder-style architecture, though much smaller than a practical one [27].

Autoencoders

Autoencoders are a very simple class of neural network which learn to take an input vector and output the same vector. That is, an ideal autoencoder simply learns the identity function:

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{3.2}$$

where $\boldsymbol{\theta}$ is trained such that $\hat{\boldsymbol{y}} \approx \boldsymbol{x}$.

The purpose of training a network like this is that in order for the network to learn the identity function it must learn two other functions: an encoding function

$$\boldsymbol{e} = E(\boldsymbol{x}) \tag{3.3}$$

and a decoding function

$$\hat{\boldsymbol{y}} = D(\boldsymbol{e}) \tag{3.4}$$

where $\boldsymbol{e}$ is the network's internal representation, *encoding*, or *embedding* of the input $\boldsymbol{x}$.

Typically, $\boldsymbol{e} \in \mathbb{R}^z$ and $\boldsymbol{x} \in \mathbb{R}^n$, where $z << n$ so that the encoding function E is a compression function, and the decoding function D is a decompression function. Figure 3.2 shows a typical (though small) autoencoder architecture.

Variational Autoencoders

While typical autoencoders learn the two functions $\boldsymbol{e} = E(\boldsymbol{x})$ and $\hat{\boldsymbol{y}} = D(\boldsymbol{e})$, *variational autoencoders*, or *VAEs* learn the two functions

$$\boldsymbol{E} = E_v(\boldsymbol{x}) \tag{3.5}$$

and

$$\hat{\boldsymbol{y}} = D_v(\boldsymbol{E}) \tag{3.6}$$

where $\boldsymbol{E}$ is a random variable drawn from a distribution which is chosen as a hyperparameter, but which is almost always a Gaussian distribution. The training goal is still for $\hat{\boldsymbol{y}}$ to equal $\boldsymbol{x}$, but since the embedding vector, $\boldsymbol{E}$ is a random variable, the VAE must learn the μ and σ which parameterize a distribution that is likely to produce an $\boldsymbol{E}$ that can be successfully decoded by D_v. This is a much more difficult task than learning the identity function, since exactly the same input $\boldsymbol{x}$ may lead to different embedding vectors, $\boldsymbol{E}$.

Why is this beneficial? The reason is twofold. First, because VAEs have a very difficult task, they must extract the most useful features from the data for compression and reconstruction. This means that VAEs are very good at feature extraction. The second reason is that VAEs are generative models. A generative model is a type of machine learning algorithm that is able to generate new data points that are sufficiently similar to ones that it has seen during training. The reason VAEs are generative is that they are forced to map their embedding space into a well-defined probability distribution (usually the normal distribution). This means that we can sample from this distribution to *generate* a new embedding vector, then feed that vector into the decoder. What comes out should be something that is new, but looks like a typical training example. VAEs have been used, for example, for human face generation by training on large numbers of photographs of faces and then sampling from the embedding space and decoding to produce images of faces not seen during training.

For this thesis, a very useful side-effect of the way VAEs work is that they tend to be quite good at clustering input data. That is, they put their embedding vectors into clusters

based on the information that is most pertinent to reconstruction of the training samples. This will be an important property for us later.

Convolutional Neural Networks

A *convolutional neural network*, or *CNN*, is a type of neural network that operates using convolutions over its input data to produce an internal representation that contains information about location-invariant patterns. CNNs are often used in image processing, where (for example) a tree may appear at any location in a 2D image, but regardless of where exactly it appears in the image, it is likely to look similar. Convolutional neural networks are more complicated than MLPs, but I make use of them in this thesis, so I explain them here in detail.

CNN architectures typically involve what are called convolutional layers and pooling layers, in addition to the normal fully connected layers. A *convolutional layer* is a three-dimensional volume of nodes, but to start with we will just concern ourselves with a single node.

Imagine a "layer" that is simply one node. This node is the input layer of the network, and so its input is a raw vector. We want this layer to perform a convolution operator over the input vector, so that it can detect position-invariant patterns. How should this be accomplished? If we know what pattern we are looking for, we can have this node do a cross correlation over the input vector [58]:

$$\boldsymbol{x}'_{\boldsymbol{i}} = \sum_{j=0}^{M-1} \boldsymbol{h_j} \boldsymbol{x_{i-j}} \tag{3.7}$$

Where $\boldsymbol{x}'_{\boldsymbol{i}}$ is the ith element of the vector that is output from this layer, $\boldsymbol{h_j}$ is the jth element of the pattern of interest (called the *kernel*), $\boldsymbol{x_{i-j}}$ is the ith element of the input vector, and M is the length of the kernel. By applying this function to each i, we can arrive at an output vector where large values correspond to a high degree of similarity between the pattern of interest and the input signal. But in most cases, we do not know all the features

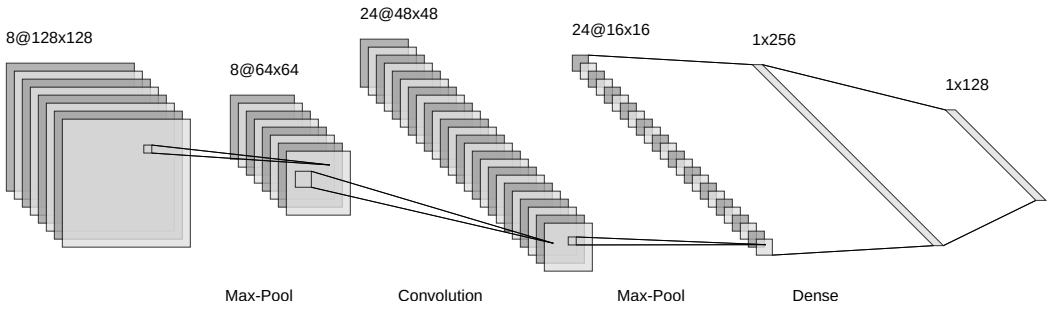

Figure 3.3: A typical Convolutional Neural Network (CNN) architecture [27].

that we are interested in - and therefore we cannot make accurate hand-crafted kernels for use in this equation. So we let the network itself learn the kernels.

To do this, we must make two adjustments to our model. First, instead of one node being responsible for the entire input signal, we make several nodes, each of which is only responsible for a segment of the input signal. Second, we associate a weight between a node and each of its input elements in the input vector. Taken together for a single node, these weights determine the kernel for that node, and they may be trained by simply using backpropagation like any other weight in the network. But rather than having each node learn its own kernel, we force all the nodes needed to cover the input signal to learn the same kernel. In other words, they all share the same weights. If, instead of a vector as input, we take a matrix (a two-dimensional signal), we could arrange our nodes into a grid that covers the entire input signal. This would be a two-dimensional slice of nodes. But each of these nodes learns the same kernel. So, to have a single convolutional "layer" learn more than one kernel, we stack two-dimensional slices, each of which is allowed to learn a different kernel. Hence, convolutional layers are three dimensional volumes of nodes which learn convolutional kernels. The output from a convolutional layer is also a three-dimensional volume, with the third dimension being the result of the convolution between a sublayer and the entirety of the input volume.

By stacking these three-dimensional layers, we have layer 0 take the input signal as its

input. Layer 1 then takes the output from the first layer. Occasionally, we downsample the signal by using something like a *max pooling layer*. This type of layer is composed of a grid of nodes, each of which takes as its input some portion of the input volume, and simply outputs the maximum value found in that volume. By alternating downsampling layers and convolutional layers, we eventually arrive at a volume that is small enough and condensed enough in features that we can reshape it into a flat vector and send it through a fully connected portion of the network to arrive at the output layer.

Figure 3.3 shows an example of a convolutional architecture. Each sheet in a layer corresponds to a kernel that is learned by the network. The sizes of the kernels (how many elements enter into each node) is a hyperparameter for each layer, as is the number of kernels for that layer. Note that although I have discussed convolutional neural networks as though they only operate on two dimensional inputs, they actually often operate on three-dimensional inputs - images with a red, green, and blue channel. But CNNs may also work in more or fewer dimensions. For example, in time-series data processing, you may want to use a CNN to learn patterns that are roughly the same shape each time, but may occur at any point in the input vector. This is perfectly plausible, and CNNs have seen such use - in fact, they have even been used in word processing tasks, where the input signal is a vector of words in a particular order. However, recurrent neural networks have historically been used more for such a task than convolutional neural networks. RNNs are what I talk about next.

Recurrent Neural Networks

A *recurrent neural network*, or *RNN*, is a type of neural network with cycles in its node network — i.e., outputs from one portion of the network are fed back in as inputs to another part. This allows it to detect patterns in signals that violate the Markov Property - i.e., signals whose ith element depends on more than just their $i-1$th. RNNs are often used in language processing, for example, where the input vector $\boldsymbol{x}$ is a sequence of characters or words. The reason an RNN is good for this task is due to *long-term dependencies*, which are simply violations of the Markov Property. For example, the tenth word in a sentence may

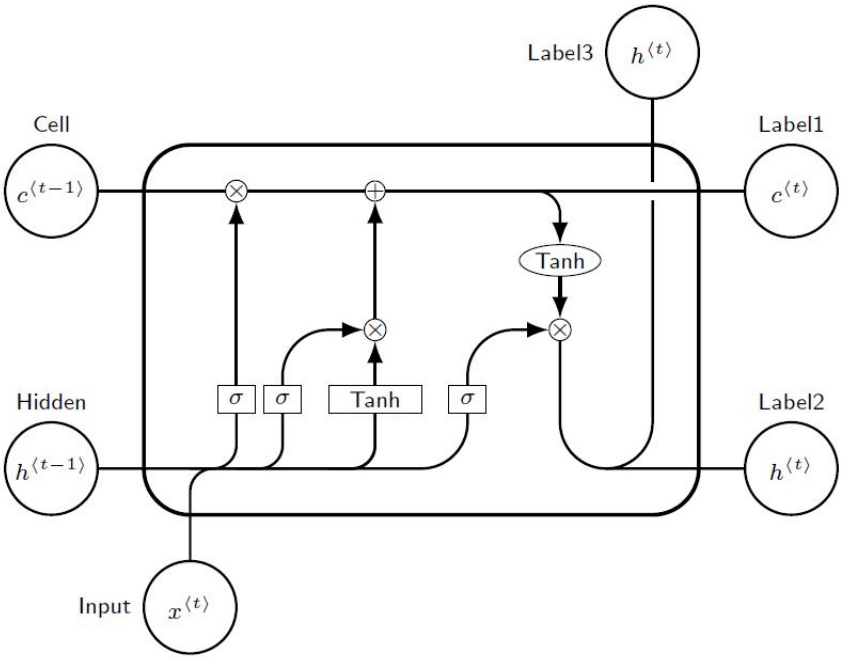

Figure 3.4: A Long Short Term Memory (LSTM) cell [28].

depend on the first, but only weakly on several in between. Naive RNNs that merely contain cycles but which are otherwise the same do not do very well in practice. Instead, a special type of RNN is typically used, called an *LSTM*, which stands for *long short term memory*.

LSTM networks are composed of many LSTM cells. Each LSTM cell looks like Figure 3.4. The flow of the diagram is this: first, the element at time point t in the input signal is fed into the cell. This element may be a vector itself. The input vector is concatenated with the hidden state from the previous time point. This concatenated vector is then fed into a normal neural network layer which uses a sigmoid activation function, which, since it is a sigmoid, will output a value in the closed interval [0.0, 1.0]. This layer is called a "forget gate", and just like any other neural network layer, it has weights coming into it that are trainable. Its output vector is element-wise multiplied by the current state c^{t-1}. Thus, each element in c^{t-1} is diminished ("forgotten") by some amount, between 0% and 100%.

Next, the input vector concatenated with the hidden state is fed through another sigmoid

layer (through different weights), and element-wise multiplied against the output of this vector after it is fed through a tanh layer. The result is then added to the cell state vector. Essentially, the previous two steps I have just described are the process of forgetting some old information and remembering some new information.

Lastly, the concatenated vector is element-wise multiplied against the output of another tanh layer, this one having as its input the modified (new) cell state. The cell and hidden states are kept and used as c^{t-1} and h^{t-1} for the next input vector. The new hidden state is passed on as the output of the LSTM cell (and may be combined with other LSTM cells' outputs or fed into an MLP, etc.).

3.1.2 Genetic Algorithms

Genetic algorithms are another type of supervised learning algorithm. Unlike typical supervised learning methods, genetic algorithms do not require that we have a dataset of *ground-truth*, or correct answers, $\boldsymbol{Y}$. Instead, genetic algorithms require merely that we have a *fitness function* that takes a guess and tells us how right it is, relative to other guesses. In traditional supervised learning algorithms, during the training phase, the algorithm produces a guess, $\hat{\boldsymbol{y}}$, which is compared against the desired output, $\boldsymbol{y}$, and an exact difference (called a *loss*) is calculated. The loss is then used to train the algorithm. With genetic algorithms however, the algorithm produces a guess, $\hat{\boldsymbol{y}}$, which is then evaluated based on some criteria. The algorithm is then told the degree of rightness or wrongness on some arbitrary scale, rather than the exact difference. This allows genetic algorithms to be used when the desired outcomes are not known ahead of time, as long as each guess, $\hat{\boldsymbol{y}}$, can be checked for correctness during training.

Like neural networks, genetic algorithms also draw inspiration from biology. Specifically, they model a population that evolves over several generations. A genetic algorithm is composed of a large number of agents, each of which may be interpreted as $\hat{\boldsymbol{y}}$, a guess at the solution. Each one is initialized via a random process, and then each one is evaluated using the fitness function to determine its correctness. The agents are then sorted based on how

well they did on the fitness function. The top n agents are then bred with one another in some fashion to generate a new gene pool, which is then evaluated again. This continues either for some predetermined number of generations, or until the fitness score has reached a certain value and stayed there. The pseudo-code for a typical genetic algorithm is given in Algorithm 2.

```
Data: G, the number of agents who compose the gene pool
Data: fitness_function
Data: N, the number of generations to cycle through
Result: F, the most correct solution found
initialize all agents in G by some random mechanism;
for n in N do
    agents_with_fvals = [ ];
    for g in G do
        fitness_of_g = fitness_function(g);
        agents_with_fvals.append(fitness_of_g);
    end
    sort(agents_with_fvals);
    F = agents_with_fvals[0];
    G = reseed_from_top_performers(agents_with_fvals);
end
```

Algorithm 2: High-level pseudocode for a genetic algorithm.

Although genetic algorithms are very useful, they are not without their disadvantages. Specifically, they are especially prone to non-deterministic behavior, as they rely heavily on randomness, and thus they come with very few guarantees about convergence. In the same vein as this is another problem: local maxima. If the fitness function produces values that have lots of "hills" (areas of high value surrounded by areas of low value), then there is very little to keep the gene pool from moving to any one of these hills and staying there. If the

hill the agents converge on is not the highest, it may not be clear, and it is probably not the solution that you want. To prevent this, it is generally best to increase the size of the gene pool to as large as possible and to produce the initial gene pool from a good sampling algorithm.

3.2 Biological Plausibility

What makes an algorithm "biologically plausible"? In this thesis, I use the term to mean any algorithm that could reasonably be replaced by a biological neural network with only minor alterations to the inputs and outputs. In this way, when I use the term "biologically plausible", I do not mean that the details of the algorithm are themselves likely to be found in the human brain, but rather I am interested in ensuring that the inputs and outputs of each algorithm are biologically sound. That is, if the algorithm were treated as a black box in the greater language-learning model, it is reasonable to postulate that such a black box may exist inside the brain - in which case it would not be implemented as a genetic algorithm for example, but rather as a biological neural network.

A reasonable question, given this use of the term "biological plausibility" is whether there are in fact algorithms that are themselves biologically realistic in their details. The answer is yes - to a certain degree. A *spiking neural network* is a type of neural network which, rather than being primarily focused on learning, is instead mostly focused on computational modeling. Spiking neural networks are constructed from neurons which are updated periodically and only fire if they reach a predetermined threshold potential. Spiking neural networks are more powerful than traditional networks because each node maintains state. But they are much more computationally complex, and machine learning libraries do not provide support for them. As such, spiking neural networks are not used in this thesis, though they are an avenue of future research. However, even if I were to use spiking neural networks instead of artificial neural networks, I would still run up against the problem of supervised learning. Specifically, artifical neural networks are trained via backpropagation, which almost certainly has no correlate in the human brain. There is some supervised learning that occurs

in biology [21] but the mechanism is not well understood.

Chapter 4

THE EVOLVING SIGNAL CHAIN THEORY OF SPEECH ACQUISITION

In this chapter, I first review the speech acquisition literature. I describe the current major theories of speech perception and language acquisition as well as studies that provide evidence for or against them. I also provide a survey of the work that has been done on computer modeling in this field. Finally, I present my theory of primary phonology acquisition, along with the empirical evidence that has led me to propose it. A computational model that was built to test key parts of this theory is introduced and analyzed in the following chapters.

4.1 Review of the Literature

4.1.1 Current Theories of Language Perception and Production

Motor Theory of Speech Perception

The motor theory of speech perception was proposed by Liberman and colleagues in 1967, but still has a small group of followers to this day. Its original version [29] asserts the following hypotheses: 1) speech is not heard in a listener as distinct phonemes - an acoustic stream is never decoded to constituent phonemes; 2) speech perception is decoded in the brain via a specialized module - sound is routed to a speech module or some other module depending on some preliminary check in the speech perception process.

The first hypothesis of this theory, that speech is never decoded into phonemes in the listener's brain, is motivated by the fact that identical phonemes may look quite different in spectrograms in different linguistic contexts. For example, the phoneme /k/ in /kaɪt/ ("kyte") does not look the same as either of the /k/s in the word /kʊki/ ("cookie"). Indeed,

later in this section I go into detail about this exact problem. However, if we are not decoding speech into phonemes, what *are* we decoding it into? The answer, according to the motor theory of speech perception, is that the internal representation is instead simply the motor commands that are used to produce the speech in the first place. That is, whatever neural command was used in your brain to issue the sound /s/ in /spikɻ/ is what should "light up" in an fMRI of my brain when I hear you utter this sound.

In fact, this is an enticing possibility for our purposes, due to it solving the so-called *correspondence problem*. When an adult produces a word /mami/ ("mommy"), and the infant attempts to produce that same sound, how does the infant know if he or she has succeeded? The sound they produce is not going to match the caregiver's sound exactly even if they articulate it in exactly the same manner. The reason is that the infant's vocal apparatus is anatomically quite different from the caregiver's. This is what gives rise to speakers having differently sounding voices. Messum et al in [35] argue that social mirroring can help with this. They theorize that infants adjust their speech-producing motor commands based on corrective signals given to them by their caregivers. When a caregiver responds to an infant's vocalization, the authors argue, the infant is able to understand this as "social mirroring" — interpreting the adult's action as doing the same thing as the infant, but correctly. To test this hypothesis, the authors created a computational model that first learns to produce non-specific speech sounds, then spontaneously utters them in "caregivers'" (participants') presence. The "caregivers" then produce a linguistically-valid utterance for their primary language. This utterance is then tied to the motor activations that the model used to produce its own utterance. In the theory presented in this thesis, I propose that infants will use the corrective signal of the caregiver when it is available to them, but when there are no caregivers around to provide such a signal, the infant will self-supervise.

Several researchers have attempted to provide evidence for or against this central tenet of the motor theory of speech perception, especially by using fMRI data [1] [13] [37] [40] [51] [57]. Unfortunately, the brain imaging data is not consistent, with some finding evidence of phonemic representations in the brain and others finding none. Most agree however, that

there is at least some motor cortex activation during speech perception, with [1] being a notable exception.

As an example of a study that does not use brain imaging [74] is informative. In this study, the authors hypothesized that if the motor theory of speech is true, then it should be more difficult for a participant to make a particular speech sound while they are also listening to speech, since the same area of motor cortex would need to be used for both tasks at the same time. More specifically, the authors hypothesized that if a participant is told to say one word while listening to another, similar one, then the word that they utter is likely to become corrupted by the distractor word. They measured participants' tongue placement using a retainer-like device. The experimental procedure was simple: participants were shown a nonce syllable that they were to say aloud, such as /sib/ ("seeb"), while listening to a similar nonce syllable, such as /tib/ ("teeb"). The researchers predicted that the participants' tongue placements would be a blend of the /s/ and /t/ phonemes (in this example). A control condition was also given, where the distractor word and the target word were the same. Indeed, the authors report statistical significance in this experiment.

However, while the above studies make it clear that humans' motor system is (often) utilized during speech perception, the central hypothesis of the motor theory of speech perception is not just that the motor system *helps* during speech decoding, but rather that it is *singularly responsible* for it. The above evidence does not seem to rise to this task. For a review of the motor theory of speech and for a dissenting opinion, see [8].

Indeed, [59] gives a fairly clear counter example: a patient whose motor cortex has been damaged to the point of being full of speech production errors, but who does not have problems with speech comprehension. I therefore fall on the negative side of this debate. I believe that the above evidence indicates that, while the motor system is useful for speech perception, it is not required.

The second hypothesis of this theory is that speech perception is handled in a fundamentally different way from other sounds. The authors of the theory believed this to be the case because it provided the easiest explanation for a few experimental findings. The most

important of these findings was that speech sounds are perceived categorically, rather than continuously. That is, people do not hear different noises when the frequency of a vowel is shifted slightly. Instead, they perceive exactly the same sound. This is useful because it allows humans to understand that when a man says "hello", it is the same word as when a woman says it, despite the sounds being quite different.

This portion of the motor theory of speech perception has been soundly disproven. For example, categorical perception in the auditory domain can be trained in humans for non-speech stimuli [36]. See [4] for additional review of this portion of the motor theory.

The motor theory of speech perception has been abandoned by most researchers in the field of speech perception and at this point it seems likely that the motor theory of speech perception's main contribution to the field of language research is that it has driven so much research into the role of the motor cortex in speech perception.

TRACE Model of Speech Perception

An alternative speech perception model is TRACE [33]. TRACE is a connectionist model — a model that invokes a neural network concept to account for language processing in the human brain. TRACE (which is not an acronym) is most easily described by a figure, as such Figure 4.1 shows the general idea.

As is clear from the figure, TRACE consists mostly of three layers: a feature layer, a phoneme layer, and a word layer. Each layer is a recurrent neural network whose outputs are excitatory traveling up the stack, but inhibitory traveling back down. For example, the phoneme layer has excitatory synapses onto the word layer, but the word layer has inhibitory synapses onto the phoneme layer. The excitatory outputs are an n-dimensional vector, where n is the number of values of that layer - for example, the phoneme layer's excitatory output is a vector over all possible phonemes in the language the model was designed for. Each element in this vector corresponds to how likely the model thinks this element (feature, phoneme, or word) is, given the information that that layer currently has.

Every t ms, buffered sound is fed into the feature layer, which will take into account

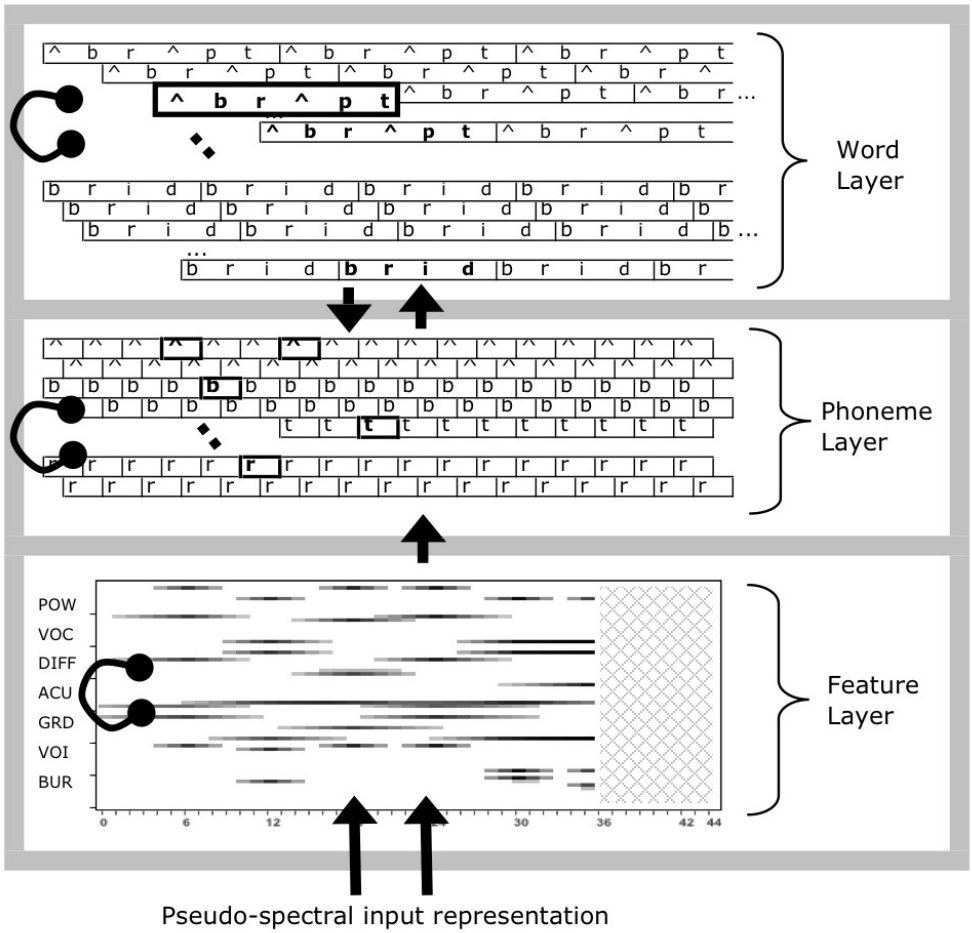

Figure 4.1: The TRACE model architecture. TRACE consists of three layers, a feature layer, a phoneme layer, and a word layer. Sound first enters the feature layer, where distinctive features are activated. The activations are then passed up to the phoneme layer, which activates phonemes. The activations of phonemes are combined with inhibitory activations from the word layer before being passed on to the word layer. This results in a word layer that can inhibit the decoding of particular sounds based on likelihoods conditioned on so-far decoded words. Each layer also contains a recurrent connection (shown on the left of each layer). (Public Domain)

information from the previous time step (from its recurrent connections) and information from the current sound (from its excitatory inputs). It will then output its excitatory output vector. This vector is fed into the phoneme layer (while another buffer of sound is being fed

into the feature layer). The phoneme layer then takes this information and combines it with its own recurrent information and its inhibitory connections. The inhibitory connections allow the word layer to inform the phoneme layer's choices in a top-down way. The phoneme layer then outputs an excitatory vector into the word layer at the same time that another buffer is being fed into the feature layer and another excitatory vector is being fed from the feature layer into the phoneme layer. The caret signs in the figure are meant to be the phoneme /ʌ/.

The TRACE model is interesting for a number of reasons. One is that it incorporates top-down processing into speech perception by letting the word layer inhibit certain phonemes if it is confident that they are unlikely, given the word so far decoded. Secondly, this model takes a connectionist approach to speech perception, which means that the authors hypothesized a biologically-plausible mechanism whereby speech is converted from sound to words. Lastly, this model was originally conceived along with a computer simulation environment that allows researchers to test the model on different (manually transcribed) utterances. This represents a very early example of using computational modeling as a tool to test hypotheses generated from a theory of speech perception or acquisition.

While this theory is interesting and has had some traction in the field of speech processing, it has not yielded experiments that can prove it one way or another. Most evidence both for and against TRACE comes from the same paradigm of studies: a subject is tasked with responding "yes" or "no" to the question "does the following word contain the sound (x)?". The subject is then presented (auditorily) with the word. As an example, let us use the sound /k/. If the presented word is "protagonist", the answer is "no" (since "protagonist" does not contain the /k/ phoneme). But if the word is the made-up word "protagokist", the answer is "yes". Part of the TRACE model of speech perception is an inhibitory connection between the lexical layer and the phoneme layer of perception. This would seem to predict that subjects should have a delayed reaction to responding in the second scenario, since the lexical competitor "protagonist" inhibits the phoneme /k/ until it is very clear to the phonemic layer and it is able to override the inhibition caused by the lexical layer. This is

a reasonable hypothesis, given the TRACE model. Unfortunately, the jury is still out, as studies have reported mixed results, with some finding this phenomenon and others failing to do so [72] [60] [49].

4.1.2 Current Theories of Primary Language Acquisition

The motor theory of speech acquisition and TRACE are both theories of speech perception as it occurs in adults. This is of interest to this thesis not only for background purposes, but also because this thesis provides a theory of phonology acquisition. If one of the above theories is true, then a theory of speech acquisition must account for it, since whatever system is responsible for bootstrapping the speech recognition system must morph that system into one of these theories. My theory falls more on the side of TRACE than on the side of the motor theory, but we shall come back to that in the next section of this chapter. This subsection describes current theories that compete with the one presented here.

Distributional Learning Models of Phonology Acquisition

Distributional learning models of phonology acquisition make the claim that infants learn the phonemes of their language by keeping track of the relative frequencies of different speech sounds, and mapping the most frequently heard speech sounds to phonemes [70]. According to DL models, infants map each speech sound into some sort of acoustic space (let's say the first formant as the x axis and second formant as the y axis). After each utterance is mapped, the infant attempts to determine the underlying distributions from which the utterances were drawn. This may amount to a simple clustering algorithm. For example, see Figure 4.2.

In this figure, the left plot shows locations of made up utterances in F1-F2 space. It is clear from this view that the utterances are drawn from two different underlying distributions. According to a distributional model of phonology acquisition, an infant would learn the plot on the right, where each cluster of utterances has been mostly enclosed by a ring. Specifically, the infant would learn the parameters that dictate the dotted red lines. After these circles

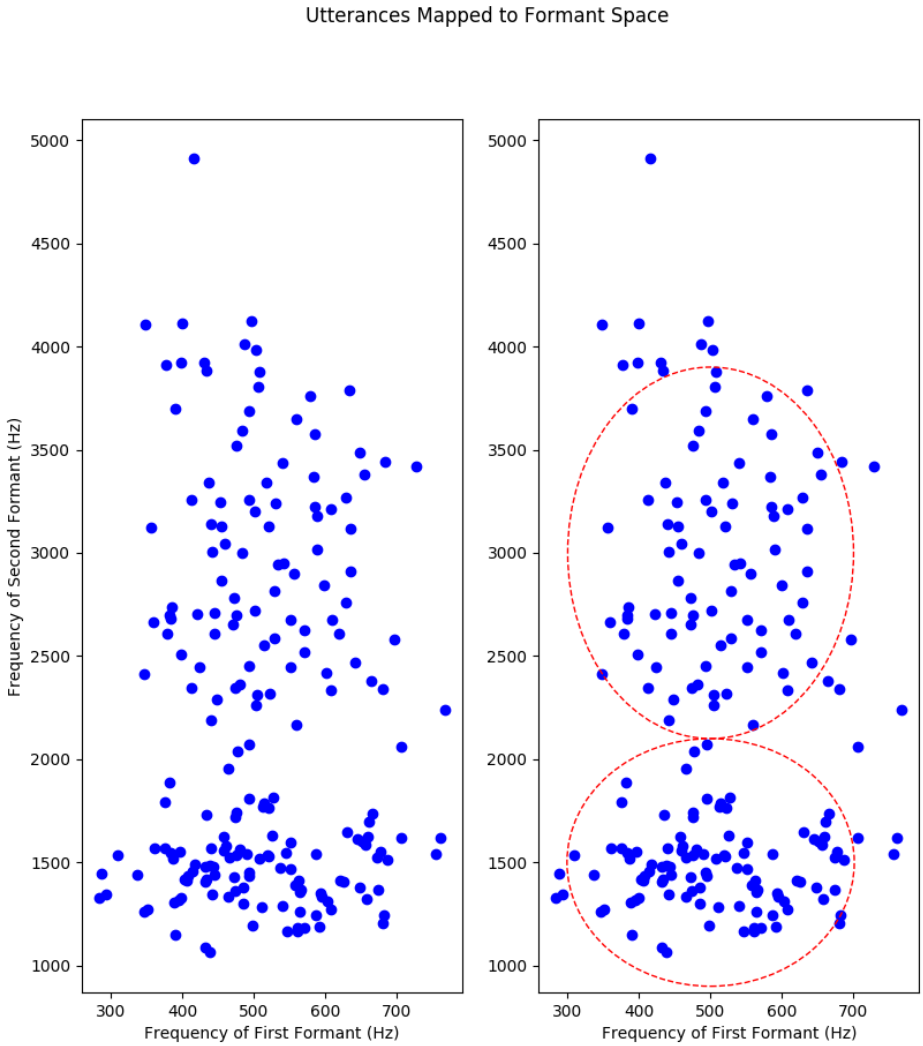

Figure 4.2: Distributional Learning. The infant must learn from the utterances it has heard (left) the underlying distributions that govern the utterances (right).

have been learned, any utterances falling into the bottom red circle would be classified by the infant as one phoneme, and any that fall into the top red circle would be classified as another.

In fact, there is experimental evidence that shows that infants can indeed do this, at least

in a contrived laboratory setting. In [32], infants were presented with /da/ sounds across the phonetic spectrum, that were still within the English /da/ category for adult English speakers. One group of infants was given a random assortment of these /da/ noises, but with the center of the distribution being in the center of the category. A second group of infants were presented instead with a bimodal distribution of /da/ noises, with each mode being close to opposite ends of the /da/ spectrum. The second group of infants was shown to be able to differentiate between the two ends of the spectrum, while the first group (the unimodal group) could not. Remarkably, the infants were only given about 2 minutes to learn the distributions.

There are however, a few issues with DL models of phonology acquisition. Most computer models and experiments that study DL as a mechanism for language acquisition use contrived examples, such as the experiment just mentioned, where the infants were exposed to only isolated /da/ sounds. But real speech is continuous and does not provide any easy way to be broken up into candidate phonemes or syllables, such as /da/. Second, there is the problem of feature space (which I shall typically call *embedding space*). Figure 4.3 (from [62], an excellent review on much of the pertinent literature) shows the problem. To obtain this graph, a mother was recorded speaking to her infant. About 700 of the vowels she used were manually parsed from the recording and then evaluated for their first and second formants as well as their duration. The top portion of the figure is a plot of these vowels in F1-F2 space. The bottom portion of the figure shows an embedding space constructed from F2 minus F1 on the y-axis, and a log scale of duration on the x-axis. The colors show the different vowel categories. No matter what two dimensional embedding space we choose for our (natural) sounds, we find a difficult task: how does an infant learn the colors?. It is entirely possible that the embedding space is more than two dimensional, but if so, it is not clear what additonal features should be used for the additional axes to solve this problem.

We should note also that formant space (which is the feature space typically used in experiments of DL) is only really amenable to vowels. Consonants (as you will see later in Figure 4.4) do not usually have obvious formants that are constant across different utterances

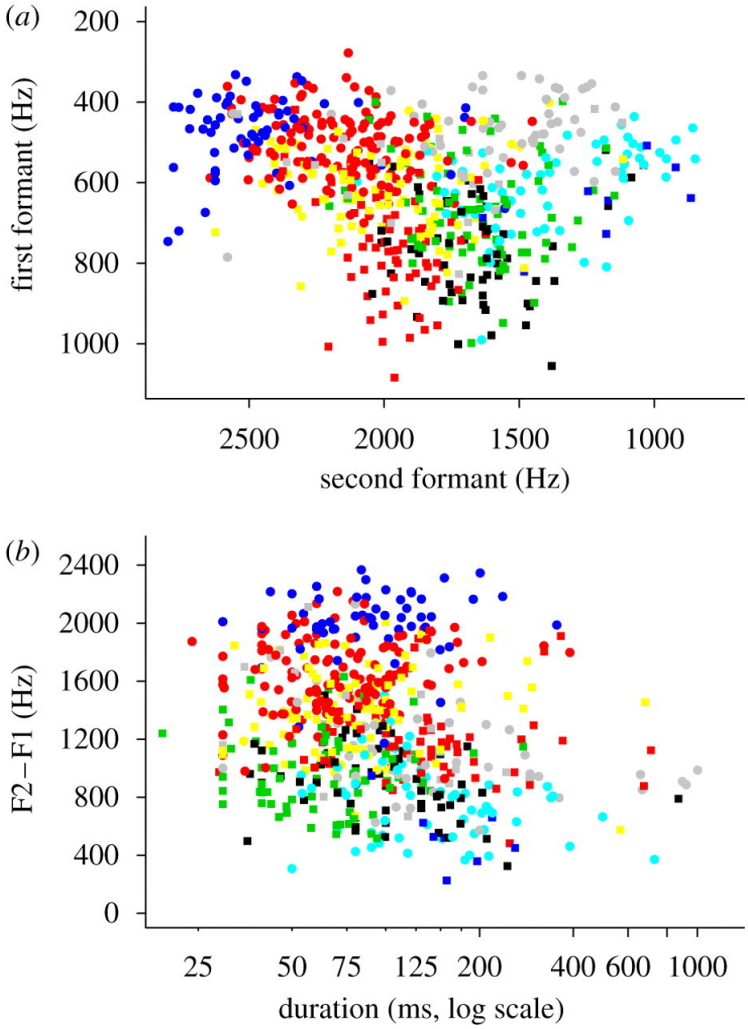

Figure 4.3: Vowels in F1-F2 space. Seven hundred vowels were parsed from a mother speaking to her infant and plotted in two different spaces. The top space is F1-F2 space, while the bottom space has F2 minus F1 on the y-axis and a log scale ms time duration on the x-axis. Infants must learn the colors, which are the different vowel categories in English [62].

— or even across different times in the same utterance.

This means that for DL to happen in an infant, two problems must be overcome. First is the segmentation problem: infants must have some way of segmenting a potentially very long utterance into a finite number of finite length vectors, each of which will be embedded in the distributional learning space. This segmenting of the speech stream will be purely

bottom up (at least to begin with, as an infant has no way of imposing structure on the speech stream yet). If gone about in a naive way, say by simply taking every 100 ms of speech, we will end up carving up similar sounds or words into different-looking vectors, which will therefore end up in very different locations in the embedding space, rather than similar ones. So DL mechanisms must have a way of overcoming this, and it turns out this task is nontrivial [62], though with a dedicated bottom-up algorithm and enough data, it is probably possible to bootstrap this process.

The second problem that DL theories face is more serious. This is the problem of embedding spaces. I defer discussion of this problem until after I cover the native language magnet theory of language acquisition.

Native Language Magnet Theory

The *native language magnet theory (NLM)* [24] [26] is a distributional learning model of phonology acquisition. This theory has three phases:

1. Infants are born with little to no linguistic ability. Infants perceive differences in speech sounds no better than non-human animals.

2. By six months, infants are able to perceive some differences between speech sounds. NLM posits that this is due to some distributional learning mechanism that embeds the utterances the infant has heard into an embedding space (the original theory says that they are "stored in memory in some form"), and then clustering them.

3. Prototypes (best exemplars) of each cluster begin to exhibit a magnetic effect on the speech sounds from now on. When an infant hears an utterance, it is segmented and the segments are placed into the embedding space, but rather than just being placed there by the DL mechanism as usual, they are mapped into a warped embedding space, where items are much more likely to fall close to one of the prototypes than far from them.

As the infant grows older, the prototypes exhibit stronger magnet effects on the embedding space, so that by the time the infant has attained adulthood, he or she has a difficult time detecting acoustic differences in utterances when those differences are not categorical (phonemic) in their native language. Additionally, NLM hypothesizes that infants learn to speak by attempting to produce the prototypes, though this portion of the theory is not as well developed.

The main point of NLM is that there exists some perceptual space wherein utterances are embedded, and that this embedding space is warped over time such that incoming points in it are much more likely to end up near specific locations (prototypes) than anywhere else.

I note here that the theory put forward as part of this thesis is a relative of NLM, but it attempts to address two main problems with NLM. First, NLM does not account for the finding that infant-directed speech (IDS) elicits vocalizations from infants, and more specifically, that infants vocalize more when caregivers imitate them [46] [7] [23]. NLM posits that IDS is important because it alters the phonetic form of speech in such a way as to make it easier to segment [26] [25] [20]. This is likely true, since it is nearly universal [25], infants will attend to it rather than adult-directed speech when given the choice [5], and it results in curious distortions of the sounds of speech [9]. Why should evolution have programmed a mechanism to distort our speech when talking to our offspring if it doesn't do anything? But there is likely more to it than just this, since otherwise why should infants produce more vocalizations during it? If IDS is mostly useful for teaching infants how to segment speech, why don't infants just listen during it? And why do adults imitate their children during these interactions? My theory answers these questions by hypothesizing that infants prefer to use IDS as a corrective signal for their spontaneous utterances, but we shall return to this in the theory section below. See [16] for a detailed review of these questions and a computational model that emphasizes this aspect of language learning.

The second, and much more important problem with NLM as I see it, is that NLM - like other theories that use DL mechanisms - fail to provide any explanation or theory of what the embedding space looks like, nor of how speech is segmented and processed before

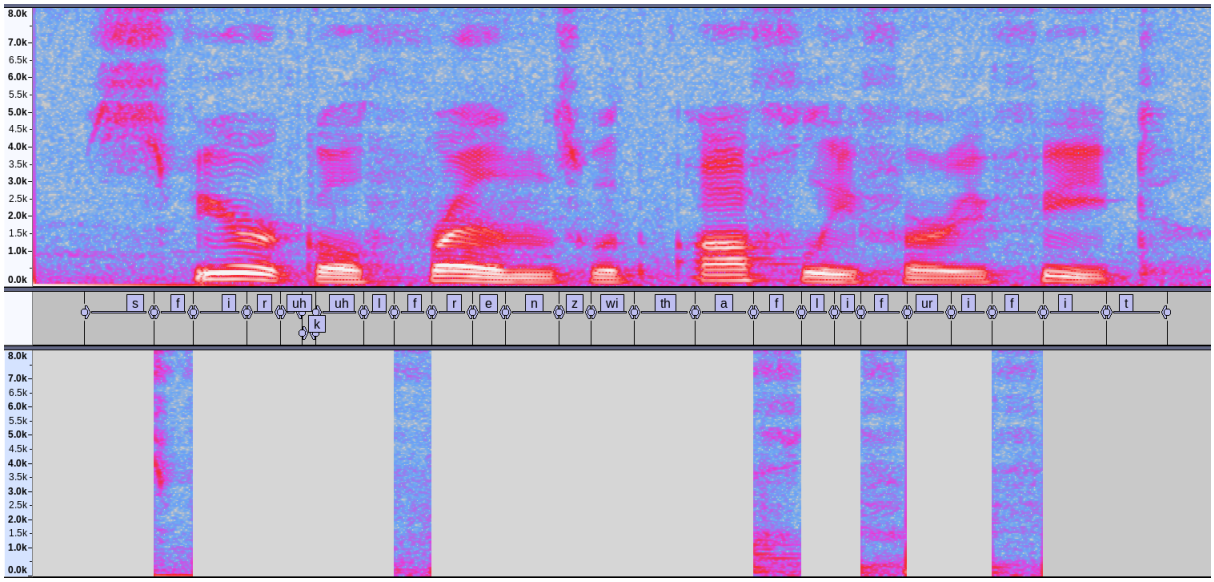

Figure 4.4: Spectrogram of an adult male saying "spherical friends with awfully furry feet", with /f/ instances parsed out and displayed on the bottom. The middle contains a rough transcription of what is being said at each time.

being embedded in it. All models and explanations of NLM that I have seen rely on hand-segmenting the input data so that it clusters nicely (such as [54] - see the computational modeling section below for further examples). But this is a chicken and egg problem. The clusters only seem to form if the infant can already parse the raw input sounds into clusterable units. But how does it learn to do this? Infant-directed speech may help with vowels, but vowels are probably the easiest to parse, as you can see for yourself by looking at any spectrogram.

Look at Figure 4.4. This is a spectrogram of me saying "spherical friends with awfully furry feet". You can see that every /f/ looks different. There is a good reason for this. They *sound* different. And if each of these /f/s sounds different, what natural embedding space exists that would place them together in clusters? I agree that reasonable embedding spaces exist that would cluster different /f/ sounds that exist in the same linguistic context, such as each time I say "feet". But the /f/ in "feet" and the /f/ in "awfully" simply don't look

or sound the same, and NLM provides no explanation for how these disparate /f/ sounds come together to form a single cluster over time. And yet they must, since adult speakers of English hear them as the same sound, even though they are demonstrably not the same. So one question that ESC tries to answer is how such an embedding space may form.

I have glossed over why these /f/s should sound different, so I shall explain it now. When you say the word "awfully", the /f/ is produced from slightly different articulator locations than the /f/ in "frown" (for example). The reason for this is that in real, continuous speech, your articulators (jaw muscles, lip muscles, tongue and supporting muscles, etc.) are moving very quickly between each sound and this leads to your muscles taking short cuts. Specifically, while your teeth are biting down on your lips to form the /f/ sound, your tongue, which is unneeded for this sound, is preparing for the next sound that it will need to be in position for. Thus, for "frown", your tongue is positioning itself into an /ɹ/ sound at the same time that you are making the /f/ sound. Meanwhile, in "awfully", your tongue is positioning itself into an /l/ sound while you are making the /f/ sound. This leads to distinct /f/ sounds - though an adult English speaker perceives them as the same sound. This phenomenon is called *coarticulation*.

PRIMIR

The Native Language Magnet model of speech acquisition supposes that infants learn language by first learning the phonemic system of their native language, and then by using that to bootstrap the learning of words. Not all language acquisition theories assume this bottom-up time course, however. There is some evidence to suppose that it may in fact be the other way around: words are learned first, and then a more parsimonious representation of them is derived, and this representation is that of phonemes. Or perhaps words and phonemes are learned at the same time and the infant uses exemplars from either category to bootstrap learning in the other. Either way, the following studies provide some evidence that words are learned before a phonemic system is fully formed: [19] found that infants as young as 7.5 months of age can segment words from continuous speech. However, infants at

about that age cannot segment words that they have been familiarized with if the speaker used to familiarize them has a very different voice from the one used to test for the segmentation [15]. Infants at 10.5 months of age do not display this speaker dependence [15]. See [68] for a more complete review of these and additional studies. These findings suggest that 8 month old infants use a simple pattern matching mechanism to parse out familiar words (a mechanism that gets tripped up when it is trained on one speaker, but tested on another), but that by the age of 10.5 months, this mechanism has become more capable of dealing with the meta-linguistic aspects of speech, such as speaker identity, stress, age, etc.

To account for these findings, Werker and Curtin have developed *a developmental framework for Processing Rich Information from Multidimensional Interactive Representations* (or *PRIMIR*) [68]. PRIMIR, like NLM, also posits embedding spaces. These spaces (in [68] called "planes") are spaces whose basis vectors are made up of different features that the infant may find useful, such as the first and second formants. Crucially, PRIMIR differs from NLM in that it supposes there are several of these embedding spaces: one for the phonetic (eventually phonemic) representation of speech, one for the lexical, and one for the meta information. Eventually, the infant adds a grammatical embedding space as well. Each of these embedding spaces takes in feature vectors derived from the speech signal and embeds them in their own spaces. Clusters form naturally, and information is shared amongst the embedding spaces in some unspecified way. In NLM, a single embedding space exists that houses proto-phonemes, which then merge over time and warp the embedding space to form phonemic boundaries. This acoustic embedding space is then used to decode the speech signal into sequential categories (phonemes), sequences of which can then be learned by higher layers as words or morphemes. In PRIMIR however, because of the existence of the lexical embedding space, words are learned at the same time as the underlying sound system. No explanation is given as to why word-parsing abilities emerge later than phoneme-parsing abilities in infants, but it is easy to imagine that this may be because forming stable clusters in the lexical embedding space simply takes longer than in the phonetic/phonemic space.

According to PRIMIR, phonemes form due to the finding of minimal pairs in the lexical

space. After enough words have been learned, there will exist enough words that differ only by one sound. These sounds must carry information then, and the sounds end up in the phoneme space as separate phonemes.

I am of the opinion that there simply is no embedding space that arises from unsupervised speech perception in which phonemes, as found in linguistic theory, will form as clusters of distinct instances of segmented sound. NLM theorizes that very few features (in computer models, often only the first two formants) are needed to get phonemes, such as /f/ to form. But as I have shown above, and as I show in the results and the discussion chapters, there is no reason to suspect this. PRIMIR adds more features, and leaves open the possibility that, even though no obvious features exist that will induce phonemes, there may be one that we haven't tested yet. It also posits that most phonemes are learned from contrasting minimal pairs. I argue that the features that must be used to put /fr/ and /fl/ and /fi/ close to one another are those that form an articulatory space. That is, the space that exists that clusters these sounds together is that of the articulatory feature space. I argue below that if phonemes exist in the brain (and they may not; see [50]), it is because they are a side-effect of the motor system, rather than of the perception system.

PRIMIR is an interesting theory of language acquisition because it makes use of all the information present in the speech signal - something that it seems likely that infants do. The main problem with PRIMIR is that it is too nebulous. It provides very few concrete ideas, especially in how the different embedding spaces interact, which is its main distinguishing feature. Any computer model that were to test this theory would need to make a lot of assumptions.

4.1.3 State of Computer Models in Language Acquisition

Computational modeling for development and testing of linguistic theories is well established. This section gives a summary of some of the more pertinent examples, especially any that test the above mentioned theories and those which I have drawn from in creating my own theory. Unfortunately, there is simply not enough space to cover here as many computational

models as I find interesting, so consider this a very brief overview. For additional review of the state of computational modeling in language acquisition, I point the reader to [53].

In 2003, the Asada group, a major player in the relatively new field of developmental robotics, put together a physical robot that could learn phonemes from humans [73]. The robot was simply a physical realization of an articulatory synthesis model - not a humanoid, mobile, or autonomous machine. The authors sought to use this model to see if vowel phonemes could arise naturally using a self-organizing map and caregiver interaction. The robot randomly controlled its articulators, and if it made a noise that sounded like a vowel, a "caregiver" (participant) would make the vowel he or she perceived back at the robot. The robot would then calculate the first four formants from this input vowel and feed them into a self-organizing map (SOM). The SOM would update using a Kohonen algorithm. The SOM was fully connected to another SOM, which was used as the controller for the articulator. The winning node(s) in the first map would then strengthen the weights that connected between those nodes and the previously activated nodes in the second map, thus making the noise that it heard tie to the activations that it had used to make a noise in the first place. The overall purpose was to move the random utterances of the robot towards the canonical utterances of the caregivers.

This represents one of the earliest demonstrations that an unsupervised learning mechanism can be used to cluster possible phonemes and to produce sounds that morph over time towards that of the sounds that the learner hears. The main problem is however, that only vowels were used, and these vowels were pure — they were not part of continuous speech. As I have discussed above, the clustering of phonemes in an embedding space (such as formant space) is only really possible if we discard suprasegmental features and coarticulation effects that arise in natural speech. Nevertheless, this is an excellent example of an early computational model of using caregiver utterances as corrective signals.

In an effort to test the key hypothesis of NLM, Salminen *et al.* used a self-organizing map to see if it would form a warped embedding space for vowels perceived via a biologically plausible model of the auditory periphery [54]. They fed artificially synthesized vowels which

varied around a mean F1/F2 location according to a Gaussian distribution into a model of the auditory periphery which led to neuron spikes in 64 different frequency bands. These spikes were time averaged to arrive at single vectors of 64 elements, each element of which represented roughly the energy of the vowel in a mel frequency band. These vectors were then fed into a SOM that used a Kohonen update algorithm. They found that the formant space did indeed warp over time, despite the variability of the input data.

While this study did a very good job of showing a concrete and thoroughly biologically plausible mechanism of the perception-space warping that NLM is based on, once again the input data was only vowels, and was not parsed from real, continuous speech.

Warlaumont *et al.* in 2013 created a computational model to study how an infant might learn to produce speech sounds that change over time without a caregiver, but with only an internalized model of how vowels sound [67]. Specifically, the researchers created a SOM that was attached to an articulatory synthesis model, which (as in [73]) was activated at random until it produced an audible sound. The sounds that were produced by the SOM/synthesizer were reinforced or not reinforced. If they were reinforced, the weights that connected the most activated nodes in the SOM to the articulators in the synthesizer were strengthened so as to make that combination of articulator activation more likely in the future. Seven different reinforcement regimens were undertaken: 1) reinforcement always given; 2) reinforcement given only if a fundamental formant could be detected (i.e., if the model produced audible noise); 3) reinforcement given if audible and similar (close in F1-F2 space) to any of the American English vowels; 4) same as (3) but using Korean vowels instead; 5) same as (3), but only for /a/; 6) same as (3), but only for /e/; 7) same as (3), but only for /u/. Reinforcement was done via the Kohonen update algorithm, as is usually the case with SOMs. The results for the /a/, /e/, and /u/ vowel reinforcement regimens can be seen in Figure 4.5.

As you can see from the figure, the SOM managed to learn to produce /a/ and /u/, but not /e/. The most important thing to take away from this is simply that this procedure, which amounts to minimizing the distance between an output sound and a target sound in an embedding space, can be used to train an articulatory synthesis model to make different

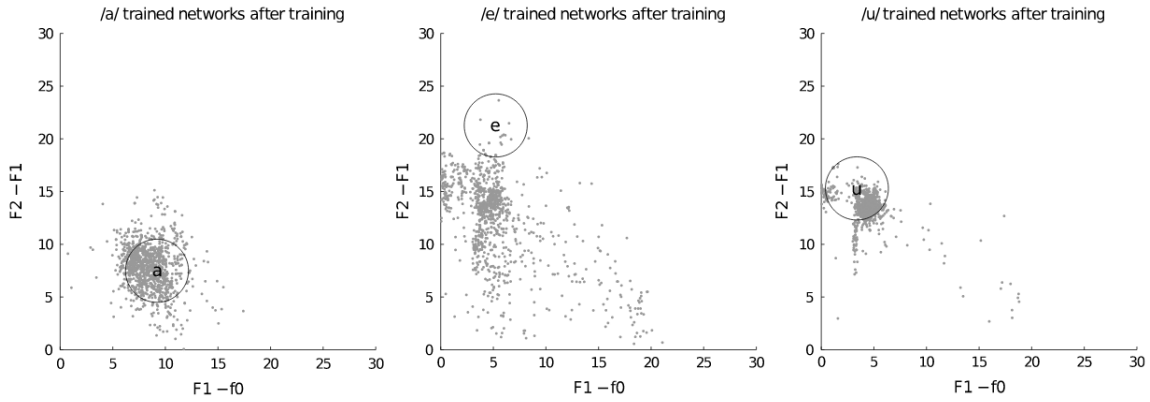

Figure 4.5: Results from [67] for the SOMs that were trained to output vowels /a/, /e/, and /u/.

noises.

Once again, this study used only vowels and F1-F2 space as the embedding space, and so the same question of how to form clusters to be used as targets arises. The targets (the circles in Figure 4.5) were known *a priori* in this case, as this study focused mostly on the production system, instead of the perception system. But the question is still valid, how do these circles get "drawn" in the human brain in the first place?

Interestingly, I cannot find any computational models of language acquisition that make use of deep neural networks. The closest I can find is a follow up study by Warlaumont that uses a spiking neural resevoir network that uses Hebbian learning and caregiver reinforcement to attempt to learn canonical babbling [66]. While fascinating from a biological plausibility point of view, this study does not introduce any of the deep learning mechanisms that have become so prevalent in the past few years. This thesis therefore may be an original contribution to the field of computational modeling of language acquisition in that it uses a deep autoencoder to learn an embedding space, rather than having one specified ahead of time. It may also be a first in that it uses an autoencoder rather than a self-organizing map.

4.2 The Evolving Signal Chain Theory of Speech Acquisition

This section covers the theory of primary phonological acquisition created for this thesis. The theory is composed of two different subsystems - speech perception and speech production. This theory combines aspects from PRIMIR and NLM to arrive at a new theory. An important aspect of this theory is that, as it has been developed, I have attempted never to stray from the biophysically and computationally realizable. That is to say that this theory contains no abstractions that do not propose a specific, computer-simulatable mechanism. To emphasize this aspect of the theory, I have dubbed it the *Evolving Signal Chain* or *ESC* theory of language acquisition, as it describes how the acoustic signal is perceived at birth, and then how it evolves over the course of language acquisition, in a concrete way, which is amenable to visualizations akin to DSP signal chain diagrams.

Before I explain the details of the theory, I first note some limitations. One, this theory currently covers only the first year of life. Additionally, this theory does not yet explicitly include anything beyond the sound system of the primary language(s). Specifically, it does not include any mention of the beginnings of lexicalization, pragmatism, semantics, syntax, or any morphology. Each of these is important, and many of them probably begin very early on in an infant's life - quite probably during the timeframe that this theory covers. Though ESC is extensible to each of these subfields, it currently does not account for them explicitly. With these limitations noted, it is time now to discuss the theory in detail.

The entire theory (shown as specified for the first six months of an infant's life) can be seen in Figure 4.6.

4.2.1 Speech Perception: Age 0 to 6 Months

Figure 4.7 shows the presented theory of perception in the first phase of an infant's life. An infant has certain in-born abilities that help to bootstrap the process of first-language acquisition. These abilities are covered in this phase. We will cover the theory by following sound as it flows from the environment into an infant's ear and into the brain.

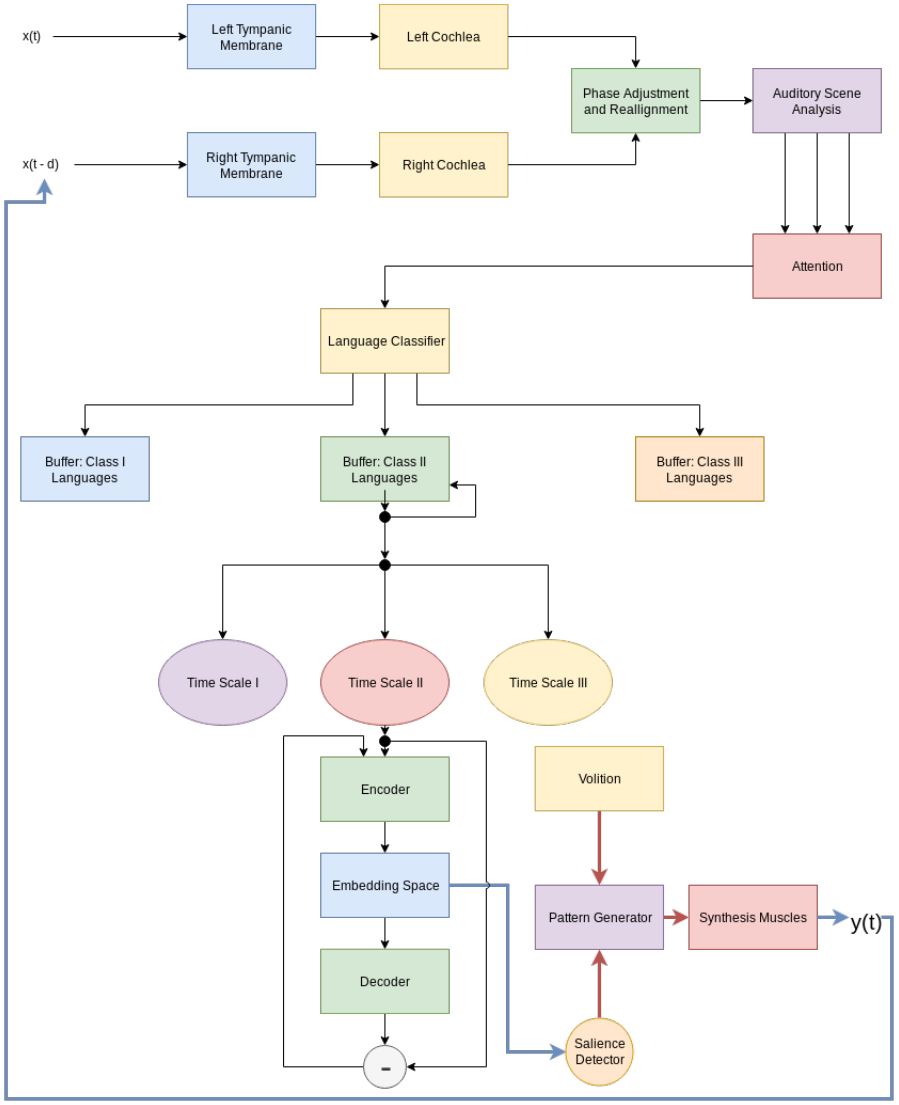

Figure 4.6: The ESC theory of speech acquisition. Sound enters at the top left in both ears and travels through the signal chain. This figure shows ESC as it is specified on the first day of an infant's life. After the infant learns to coo, the saliency detector (at the bottom) is replaced with a comparison block as shown later. Sounds output by the infant are fed back in, since the infant can hear sounds that it makes.

Sound enters the infant's left and right ears at slightly different times, depending on the exact location of the sound source in relation to each ear. This results in sounds that are phase-shifted versions of one another entering the brain. Before they enter the brain

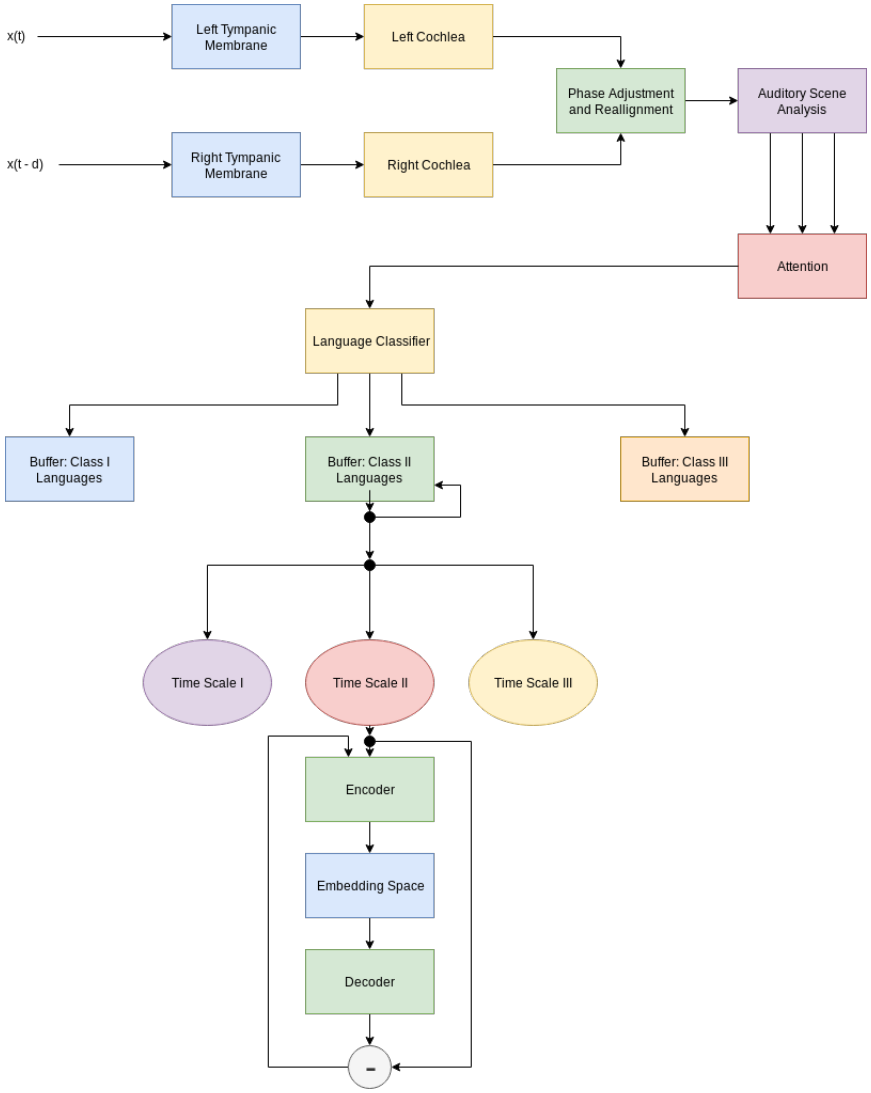

Figure 4.7: ESC: Perception. Only the perception portion is shown. This portion of the theory changes over time mostly by training the autoencoders (only one of which is shown) and by removing language classes that do not receive input.

however, they first pass through the tympanic membrane of each ear, which has the effect of changing the medium of conduction from air to bone (by means of the auditory ossicles). The ossicles then transmit the time-domain vibrations to the oval window, where the medium of conduction is changed once more, this time to the fluid inside the cochlea.

The cochlea is composed of a basilar membrane, which has a thickness that changes from

thicker to thinner as sound travels down the cochlea. This means that the basilar membrane vibrates at different locations according to the frequency components of the sound. Sitting atop the basilar membrane is a specialized group of cells called *hair cells*, which have long hair-like cilia on one end. As the basilar membrane vibrates, it displaces the hair cells such that they may push their cilia into a membrane that sits above them, called the tectorial membrane. This physical displacement of the hairs on these cells causes depolarization and nerve impulse conduction to the brain. In this way, the cochlea acts to convert the time-domain auditory signal to the frequency domain.

The impulses travel along different frequency-sensitive neuron bundles to the brain stem and then up through the thalamus and into the neocortex [47]. Here, most of the auditory processing that we shall be interested in takes place (specifically in the temporal lobe). At some point, the two phase-shifted sounds are integrated into a single sound [31].

At this point, the theory enters a more conjectural portion, as the neuroanatomy and psychology of these processes enters the realm of the theoretical. The single source sound enters an *auditory scene analysis* (ASA) block. This block takes the sound and multiplexes it into different channels based on different sources. This block is responsible for an infant being able to tell the difference between speech, background hum, and whatever else - though to begin with, an infant may only be able to tell the difference between sounds based on a few criteria.

These different channels travel to the next block, a sound attendance mechanism. This mechanism determines which channel of sound the infant should pay attention to. Infants can discern several different auditory stimuli at birth and shortly after birth, inluding their mother's voice [34]. While it is very difficult to determine what sound an infant *prefers*, several researchers have made efforts to do just this [64] [5]. It is my belief that the language of these studies (do neonates really have things that they *prefer* or *like*?) subjects them to bias. Nonetheless, infants pay attention to certain sounds and less so to others, and they can tell the difference between speech and nonspeech [64]. This block determines the channel the infant should attend to, and if that channel is speech, it continues into the rest of the

model. If it is non-speech, it may go elsewhere, but possibly it enters the rest of the model and simply fails to elicit responses. In either case, we do not consider the case of attendence to non-speech acoustic stimuli in this theory.

A few studies have shown that infants are actually able to tell the difference between broadly dissimilar languages, even at birth [41] [38]. A block is introduced in the theory, therefore, at this point to route the speech to an appropiate language-processing network. I hypothesize based on the results in these studies that the human brain begins with a small number of these modules - perhaps three (as shown in the figure). The detection mechanism in newborns is primitive and probably rhythm-based [41] [38]. (Interestingly, this is also true for monkeys [52]). Each module will decay over time if not used, or else it will grow, change, and diversify in the coming months.

Once the sound enters one of these language-family-specific modules, it is examined by something like a convolutional network at several different time scales. The longer time scales attempt to learn the prosody of the language and whole words (especially function words and frequently occurring shorter words), while the shorter time scales are seeking to learn the phonemic combinations of the language. Notice that I do not say "phonemes". Instead, this network learns an embedding space that contains clusters of sounds that are similar - so that different utterances of /f^{r}/ end up close to one another and different utterances of /f^{l}/ end up close to one another, but /f^{r}/ and /f^{l}/ may not necessarily end up close to one another. In the figure, I have shown this mechanism as it is implemented in the reference system (described in the next chapter), as an autoencoder, but it need not be - it could be a self-organizing map, for example, so long as the embedding space is learned rather than specified *a priori*.

4.2.2 Speech Production: Age 0 to 6 Weeks

While the embedding spaces are being learned in the perception portion of the model, the child is learning to coo with the production portion of the model. Figure 4.8 shows what is happening in the child's brain during the first six weeks of life insofar as speech production

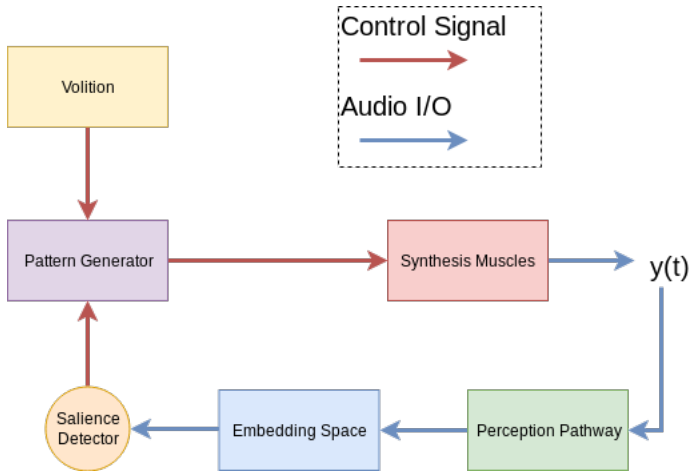

Figure 4.8: ESC: Learning to coo. The green perception pathway block represents all of Figure 4.7 except the embedding space block, which is explicitly called out here.

is concerned. A child is merely learning the muscle control needed to accurately utilize the various articulators. The goal of this phase is to learn to vocalize. The accuracy of the vocalizations in regards to the linguistic environment is mostly ignored during this phase, with vowels being the first sounds made, since they are simply the easiest to get the articulators to produce. The process by which the infant learns to control their articulators is governed by intrinsic motivation - that is, children are simply hard-wired to do this at this age, as they "enjoy" it. This is similar to how toddlers find joy in practicing motor skills that they will find useful later in life - such as climbing stairs [65] [39].

How does an infant know if it is vocalizing correctly? Again, this phase is merely about vocalizing *at all* rather than "correctly". Also, deaf children go through this phase as well [43]. This means that most likely, infants learn to control their articulators based mostly on proprioceptive feedback, though I would guess that hearing children also use the sound that they produce (once they are able to produce sound) as a reward signal as well.

This process is captured by Figure 4.8. First, a volitional block supplies a random signal to a pattern generator. This volitional block is kept intentionally vague, as it is the only place the theory currently interacts with higher level cognitive processes, and I do not wish

to specify these processes, as it is clearly outside the scope of this work (why does an infant choose one sound or another? Why does an infant choose one word or another? These questions are not to be answered by this theory at this time, though see [39] for a model that emphasizes precisely this aspect of language learning). Thus, in a reference system, this block requires the implementer to make decisions that are not specified here. Although I have made every effort in developing this theory to keep it as concrete as possible, some amount of vagueness is necessary, as we simply do not know enough about the human brain to specify all the interfaces for a proposed system such as this.

The pattern generator outputs a random control vector every some number of milliseconds, which enters the motor cortex and creates nerve impulses that specify movement of articulator muscles. These movements, in combination with the action of the lungs at the right time, lead to an output sound signal being created. This sound enters the infant's ear and travels through the perception system. It is possible that this sound is short-cutted to the right language classifier and that other optimizations may occur on the infant's self-produced sound, but I do not choose to specify anything in this vein at this time. Either way, the sound's embeddings are used to adjust the pattern generator to make vocalizations in the future more likely.

4.2.3 Speech Production: Age 6 Weeks to 6 Months

Figure 4.9 shows the second stage of speech production. During this time, the infant is learning to produce reduplicated babbling, such as /bababa/. This stage is governed by the child learning to utilize the articulators with more expertise and then, after a time, it is governed by the connection of this motor system to a recurrent network (not shown). By the time this phase of the model has been entered, the embedding spaces have been formed and sounds (at least in the shorter time windows) are embedded in these spaces deterministically.

Figure 4.9 shows the volitional block sampling sound from the embedding spaces found in the perception portion of the model, then feeding some information derived from that embedding into the pattern generator. The pattern generator then issues a control vector to

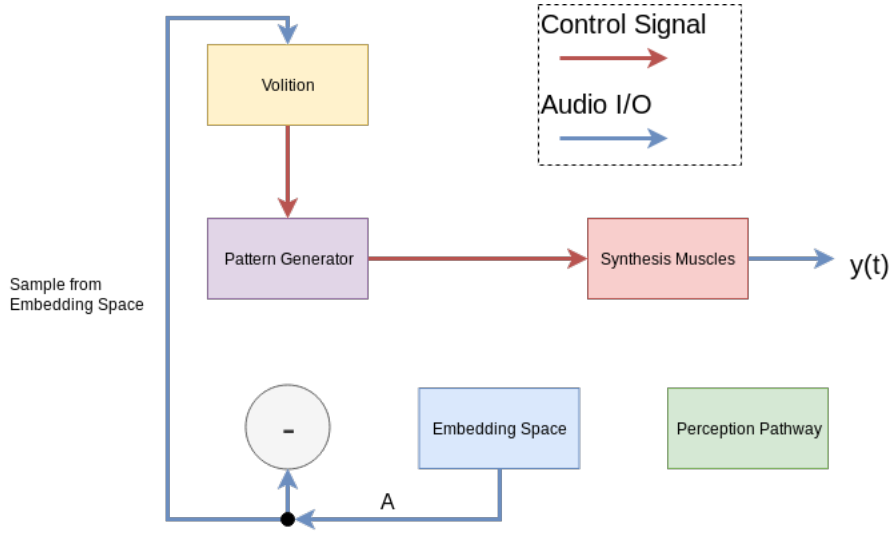

Figure 4.9: ESC: Learning to babble (A). After the infant has learned to coo successfully and the autoencoders have been trained, it moves on to learning to babble. Sound is sampled from one of the trained embedding spaces (A) and fed into the volitional block, which then attempts to get the production system to produce this sound. Compare with Figure 4.10.

the muscular system, which produces a sound.

Figure 4.10 shows this sound being fed back into the perception system and ultimately back into the embedding space(s). The location of this sound in the embedding space (labeled B) is compared against the location of the target sound (A). This comparison is used to adjust the pattern generator so that next time A is fed into it, it will produce a sound that ends up closer to B in the embedding space from which it was drawn.

Much research has assumed that infants self-supervise while learning to produce speech. That is, an infant will attempt to produce a sound, then hear the sound they actually produced and compare the target with the actual internally. The comparison will then lead to some internal changes that correct for the perceived error. In this way, infants may reduce the difficult task of unsupervisedly learning to produce speech to a more tractable supervised process. In this thesis, as I have just explained, I agree that some amount of this is present. But I think it is unlikely to be the entirety of how infants learn to produce

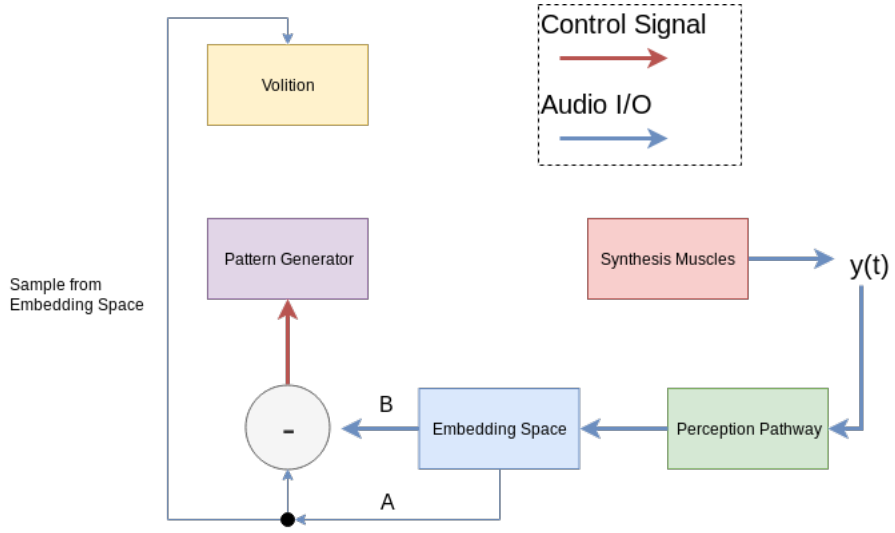

Figure 4.10: ESC: Learning to babble (B). When sound is made by the infant, it hears the sound, which travels down the perception pathway and into the embedding space from which A was sampled. The locations of the sounds are compared in the embedding space, and the production system is modified such that the locations will be closer together in the future. Compare with Figure 4.9.

viable speech signals. One reason for this is the above-mentioned correspondence problem. The second reason is the observation that caregivers mimic their children much more than vice versa [22]. Thus, this theory takes the view that infants self-supervise when the better caregiver-provided corrective data is not present, but when caregivers are present, infants prefer to take the corrective signal from their adult counterparts and ignore A in the figure.

4.2.4 Combined Perception and Production: Age 6 to 11 Months

In the final phase of this theory, the perception and production systems are fully combined into a single speech system. This system connects the embedding spaces with the muscle commands. The infant then tries to produce the speech sounds that it has heard. The reward signal that trains it is based on how well the produced sounds match the sounds it hears in the environment. If a caregiver is present and can provide an instructive signal, this signal is used instead of the default training behavior, and it is given more weight. In deaf children,

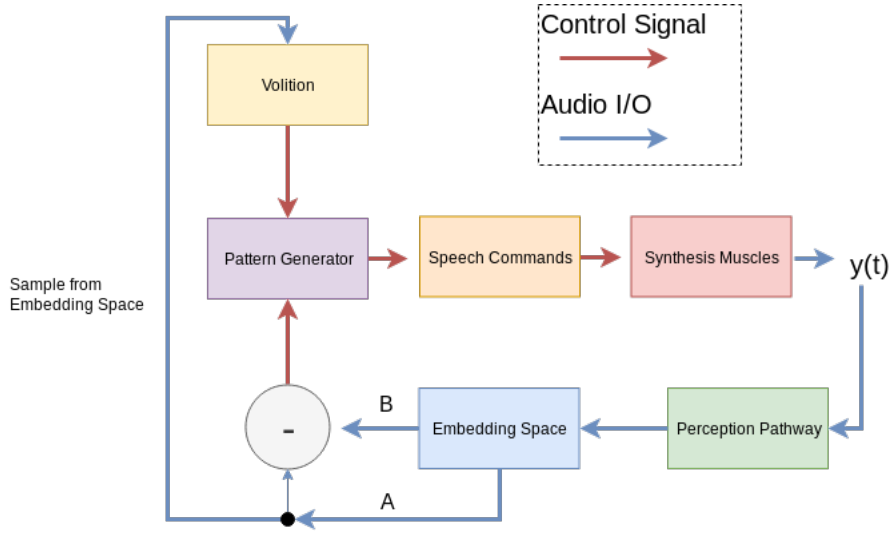

Figure 4.11: ESC: Learning speech commands. By the time marginal babbling has been mastered by the infant, the infant begins to extract motor commands into a hierarchical representation, which will eventually form a link with the embedding spaces to help in speech perception. In this way, the motor system is responsible for "phonemes".

this phase is co-opted by the visual and proprioceptive systems in some manner. But in hearing children, a byproduct of this phase is the production of canonical (nonreduplicated) babbling.

Eventually, the control of the articulators is handled by a hierarchical control system, with higher layers producing sound prototypes whose details are filled in by lower layers, as shown in Figure 4.11. This is how the muscles are controlled in the human body — when you command your arm to move, a high level signal is generated that leads to lower level neurons actually manipulating the muscles and providing stability and postural control to carry out the command. In the case of producing speech, the high level commands are learned after the low level details of sound production are already well practiced. The theory of ESC is somewhat agnostic to the existence of phonemes *per se*, but if phonemes exist in the brain, the theory posits that it is here — as high level speech production *commands*, rather than as abstract symbols that are used in the decoding of a speech stream.

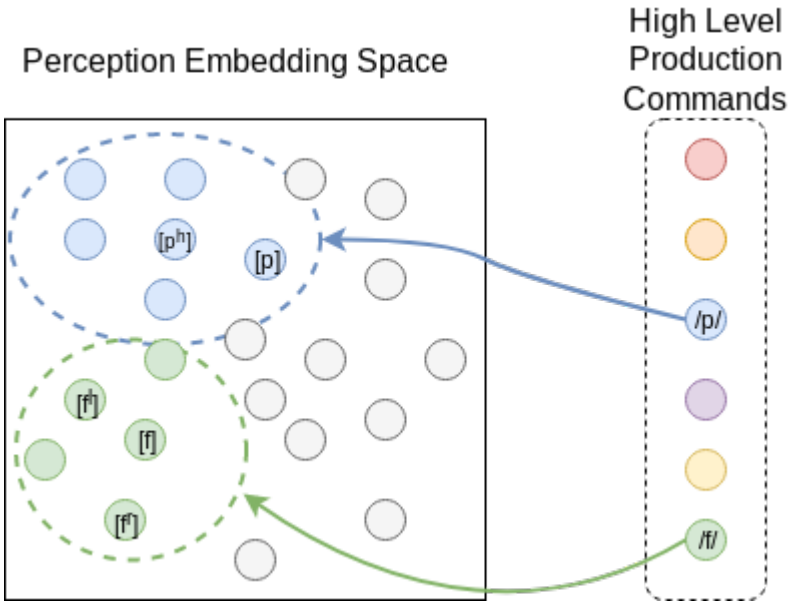

Figure 4.12: Categorical Perception in ESC. Eventually, the high level speech commands form a connection to the embedding spaces. This connection is Hebbian: when [p^h] is heard, or when [p] is heard, both light up /p/ in the motor system.

These high level speech production commands (themselves making up an embedding space of sorts) eventually associate with the perception pathway's embedding spaces. Sounds that are similar to one another will, by the nature of the unsupervised learning algorithm used to learn the embedding spaces, end up close to one another. These similar sounds associate with high level speech production commands as showin in Figure4.12. This is how ESC accounts for categorical perception.

Chapter 5

METHODS

The previous chapter discussed a new theory of primary phonology acquisition. One of its basic tenets is that it be, at least in principle, concrete enough that it is implementable as a computer simulation. This chapter discusses the efforts I have made to create a reference implementation of ESC, as well as the steps I have taken to use it to test the theory.

This chapter is composed of three parts. First, I describe the model that I have so far implemented. It is not a complete reference implementation yet, but most of the key features of the theory are done. The second part of this chapter describes the data I have collected to test this model. The third part describes the experiments I have run on the model in an effort to test the theory of phonology acquisition as presented in this thesis. The next chapter describes the results of these experiments, and the chapter after that discusses the implications of these results.

5.1 Reference Model

This section describes the reference implementation of the Evolving Signal Chain theory of phonology acquisition. For the remainder of this chapter, I will typically refer to this reference implementation as simply "the model". This will be distinguished from the abstract theory as presented in the previous chapter, by calling the more abstract, underlying model "the theory".

I first describe the model in a way very similar to how I described the theory in the previous chapter, by walking through the signal chain and the time course. Wherever assumptions have been made or the model deviates from the theory, I describe in detail the reasoning for these choices. Next, I describe the model in more detail as a software system - I describe

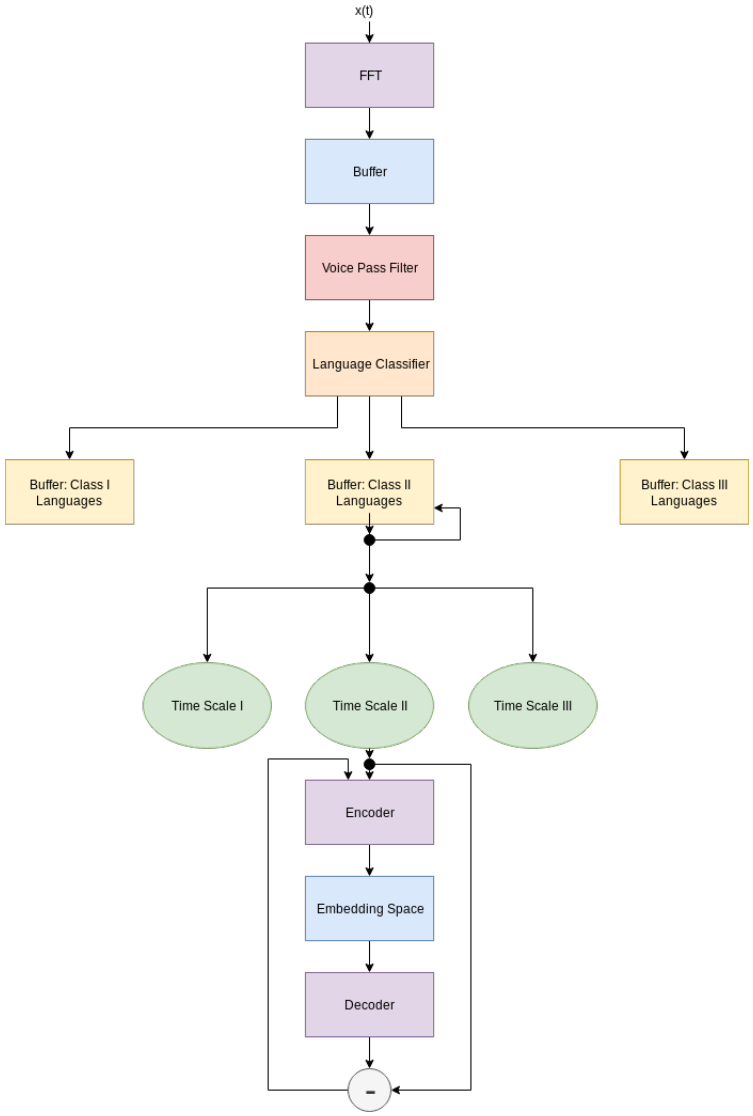

Figure 5.1: ESC Perception Reference Model. Sound enters at the top and goes down.

the programming language, the software architecture, and the general design choices that I have made, and the reasoning behind these choices.

5.1.1 The Model

The perception portion of the ESC reference system is presented in Figure 5.1. In principle, this reference system could be made to run in real time on an embedded robotics system. Indeed, this is how it was originally envisioned. But in the interest of time, engineering efforts for this thesis were directed towards testing key parts of the underlying theory rather than in the optimizations that such an endeavor would require. Future work may focus on a robotic platform; for a discussion, see the conclusion chapter of this work. In any case, I will describe the model here as if it were in a real time system.

Raw, unlabeled sound is fed into the system. First, the sound is converted into the frequency domain by means of a fast Fourier transform (FFT). The frequency domain signal is then fed into a buffering block that outputs spectrograms of a long time frame. This differs from the underlying theory in that it is a very rough approximation of the auditory periphery. For example, it does not assume binaural audio and as such makes no effort to correct for the resultant phase shift. This is, as far as the theory goes, merely a simplification, and should make no difference to speech acquisition.

The next block in the signal chain is a voice activity detector, which I have labeled in the figure as a voice pass filter to emphasize that it serves merely to filter all sound that is not speech. This also differs from the underlying theory in a manner that serves to simplify preliminary processing of the audio data. The theory demands an auditory scene analysis block, which multiplexes the sound signal into different channels based on sound source. Such a signal block is possible: see [17] for a bottom-up (and therefore biologically plausible in infants) auditory scene analysis block that could be used. The main disadvantage of removing the ASA block and replacing it with a voice pass filter is that it removes flexibility from the system. The voice pass filter acts as a combined ASA/attention block that is hard-coded to multiplex the audio into voice/non-voice channels and then to discard the non-voice channel. With true ASA and attention blocks, we could separate overlapping voice data into lower signal quality channels of single voice each. Then we could choose to attend to the

channel with higher signal quality. Or we could choose to attend to a more familiar voice, as infants have been shown to do with their mothers [34], or to a voice that is engaging in infant-directed speech.

The sound then enters a language classifier block which, for each spectrogram entering it, assigns the spectrogram to one of three different possible language categories. The details of this block are unimportant for this work because, although the block is part of the reference system, it was not used. The rest of the perception portion of the model is the same as the theory.

Two implementation details should be noted here. First, although the reference system calls for a voice pass filter, and one was implemented, in the experiments that are reported for this thesis, it was not used. The reason for this was that it segmented the input sounds too much, which resulted in odd discontinuities in the time domain, thus leading to artifacts in the frequency domain. Due to time constraints, I opted to simply use a silence filter instead. The consequence of this is that the autoencoder was faced with a more difficult task: to encode all sound rather than just voice. The second implementation detail that diverges from the reference specification is that no language classifier was used. Again, one was implemented and tested, but its results were similar to the voice pass filter, and so it was removed from the architecture. Again, the consequence of this is that the autoencoder is faced with a more difficult task: to embed sounds from any language that is found in the dataset. For the dataset used in this experiment, this meant that Mandarin and English both ended up in the same embedding spaces. For the particular experiments done for this thesis, these differences should not matter.

Although this system was envisioned as a real time system, it is currently implemented as a batch processing one. Raw, continuous data was recorded, then it was loaded onto a hard drive and fed into the system. First, the sound files (which were of varying lengths) were silence-filtered and the results were batched into ten minute long audio files, sampled at 32 kHz, 16-bit mono, uncompressed (WAV) format. After this, the batched audio was converted to two different groups of spectrograms. Since the theory calls for segmenting speech in at

least two different time scales, two time scales were used for the spectrograms: one group was therefore downsampled to 16 kHz, 16-bit mono, 0.5 second long spectrograms windowed by means of a Tukey windowing function ($\alpha = 0.5$) at 0.03 seconds per window, with a 20% overlap between each window; the other batch was downsampled to 8 kHz, 16-bit mono, 0.3 second long spectrograms windowed by means of a Tukey windowing function ($\alpha = 0.5$) at 0.02 seconds per window, with a 20% overlap between each window. The first group was composed therefore of spectrograms of dimensionality 241 frequency bins spanning 0 to 8 kHz by 20 time bins spanning 0 to 0.5 seconds, and the second group was composed of spectrograms of dimensionality 81 frequency bins spanning 0 to 4 kHz by 18 time bins spanning 0 to 0.3 seconds. Note that these spectrogram frequency bins are linearly spaced within their ranges, which is in contrast to the mammalian (including human) auditory system, which follows a more logarithmic relationship. Secondly, note that the preprocessing of the sounds was done in reverse order from Figure 5.1: silence filtering was done first rather than spectrograms. This is purely an implementation detail, as it was easiest to filter silence in the time domain. Lastly, note that although the figure shows three time scales, only the above mentioned two were used in the experiments that follow.

The spectrograms were fed into one of two deep convolutional autoencoders (see Tables 5.5, 5.6, 5.7, and 5.8), depending on the spectrogram dimensionality.

Next, I describe the reference implementation of the speech production portion. Figure 5.2 shows the production system while it is learning to coo. This is the same figure as Figure 4.8. Some mechanism outputs a random signal that is input into a pattern generator. This pattern generator creates articulator control vectors, which are then fed into the articulator system, which may output a noise. If there is noise, this is fed back into the perception system, into the embedding spaces, and then used somehow in a salience detector. The output of the salience detector determines how the pattern generator should change over the course of learning to coo.

In the experiments done to test the model's ability to learn to coo, the volitional block is simply a "for loop" that stimulates the pattern generator some number of times before we

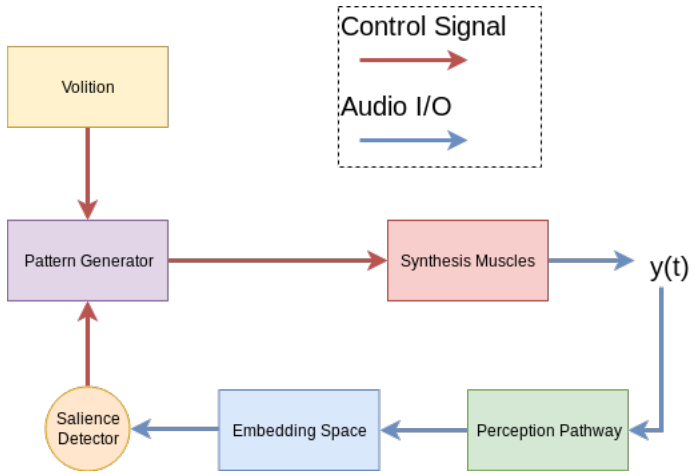

Figure 5.2: ESC Production Reference Model: Cooing. This is the same figure as Figure 4.8.

move on to learning to babble. The pattern generator is a genetic algorithm that outputs control vectors based on random search guided by a fitness function. The control vectors are fed into Praat's articulatory synthesis mechanism [3], and the output of the synthesis is then evaluated for its root mean square (RMS), which is directly used as the fitness function for modifying the gene pool in the pattern generator.

The Praat control vectors are matrices with rows corresponding to articulator muscles and columns corresponding to time points. Each element in the matrix is an activation for a particular articulator. These values are linearly interpolated between time points.

To run the synthesizer, a user must supply a matrix of control values, each of which represents the "activation" of a particular articulator at a particular time. Table 5.1 gives an example of one of these matrices. This matrix produces a sound like [əbwə], when the total duration of the produced sound is set to half a second and where t_0 is at 0 seconds, t_1 is at 0.10 seconds, t_2 is at 0.25 seconds, and t_3 is at 0.5 seconds. This example is taken from Praat's documentation.

In order to produce an output sound, one of these matrices must be fed to the articulatory synthesis model. The articulatory synthesis model is deterministic, and so the same matrix will produce the same sound every time. Thus, in order to produce different output sounds

Table 5.1: Example articulatory synthesis control matrix. Non-zero values are bolded.

Articulator	**Value at t0**	**Value at t1**	**Value at t2**	**Value at t3**
Lungs	**0.2**	0.0	0.0	0.0
Interarytenoid	**0.5**	**0.5**	**0.5**	**0.5**
Cricothyroid	0.0	0.0	0.0	0.0
Vocalis	0.0	0.0	0.0	0.0
Thyroarytenoid	0.0	0.0	0.0	0.0
PosteriorCricoarytenoid	0.0	0.0	0.0	0.0
LateralCricoarytenoid	0.0	0.0	0.0	0.0
Stylohyoid	0.0	0.0	0.0	0.0
Thyropharyngeus	0.0	0.0	0.0	0.0
LowerConstrictor	0.0	0.0	0.0	0.0
MiddleConstrictor	0.0	0.0	0.0	0.0
UpperConstrictor	0.0	0.0	0.0	0.0
Sphincter	0.0	0.0	0.0	0.0
Hyoglossus	0.0	0.0	0.0	0.0
Styloglossus	0.0	0.0	0.0	0.0
Genioglossus	0.0	0.0	0.0	0.0
UpperTongue	0.0	0.0	0.0	0.0
LowerTongue	0.0	0.0	0.0	0.0
TransverseTongue	0.0	0.0	0.0	0.0
VerticalTongue	0.0	0.0	0.0	0.0
Risorius	0.0	0.0	0.0	0.0
OrbicularisOris	0.0	0.0	**0.2**	0.0
LevatorPalatini	**1.0**	**1.0**	**1.0**	**1.0**
TensorPalatini	0.0	0.0	0.0	0.0
Masseter	0.0	0.0	**0.7**	0.0
Mylohyoid	0.0	0.0	0.0	0.0
LateralPterygoid	0.0	0.0	0.0	0.0
Buccinator	0.0	0.0	0.0	0.0

over time, the matrices must change over time. To provide a mechanism by which this could be done, I implemented a genetic algorithm which outputs control matrices.

The genetic algorithm works as shown in Listing 3.

```
Data: G, the number of control matrices in the gene pool
Data: fitness_function
Data: N, the number of generations
Data: E, the number of matrices to breed in each generation
Data: X, the number of matrices to mutate in each generation
Result: M, the best control matrix found
genepool = seedfunction(G);
for generation_idx in range(N) do
    /* Compute the fitness of the genepool                    */
    fitnesses = evaluate_fitnesses(fitness_function, genepool);
    sorted_fitnesses = sort(fitnesses);
    /* Sort the genepool based on calculated fitnesses        */
    sorted_genepool = sort(genepool, sorted_fitnesses);
    if generation_idx == N - 1 then
        /* If this is the last generation, return the best matrix */
        return sorted_genepool[0];
    end
    /* Take the top E matrices from the genepool              */
    breeding_pairs = take_top_matrices(sorted_genepool, E);
    /* Breed the top E matrices using crossover               */
    genepool = crossover(breeding_pairs);
    /* Mutate X matrices                                      */
    genepool = mutate(genepool);
end
```

Algorithm 3: Pseudo code for the genetic algorithm

The genetic algorithm starts with randomized control matrices, and at each generation,

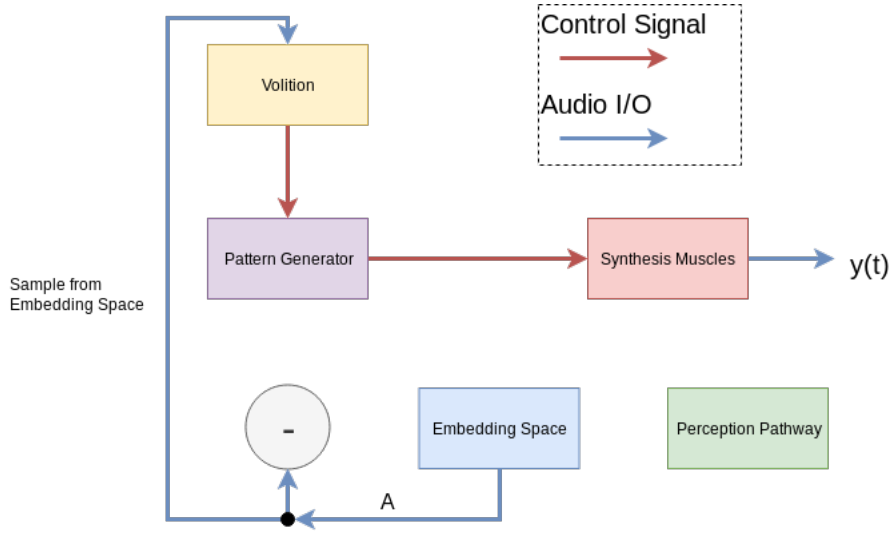

Figure 5.3: ESC Production Reference Model: Learning to babble. This figure is identical to Figure 4.9.

tests each one. The best ones are kept and mixed with one another in a cross-over procedure to produce the next generation. Some of this generation is then mutated to introduce more variance into the search. In order to test the control matrices, fitness functions were developed that took the sounds those matrices produced and tested them to output a fitness score. In this phase of the model, RMS is used as the fitness function. Thus, it should be the case that over time, the model produces more utterances with larger RMS, which should correspond to learning to make noise (which I hypothesize is the same mechanism by which infants learn to coo - they simply learn to make noise, and the first noises they make are the easiest ones to learn, which come out as coos).

After learning to coo, the model moves on to learning to babble. This phase is shown in Figures 5.3 and 5.4. These are the same figures as Figures 4.9 and 4.10. The volition block chooses some regimen for training the pattern generator, samples an embedding from the embedding spaces, and inputs a signal into the generator based on these inputs. The pattern generator outputs a control vector which is fed into the synthesizer, which may output a noise. The noise is fed back into the system, and its embedding(s) is/are compared

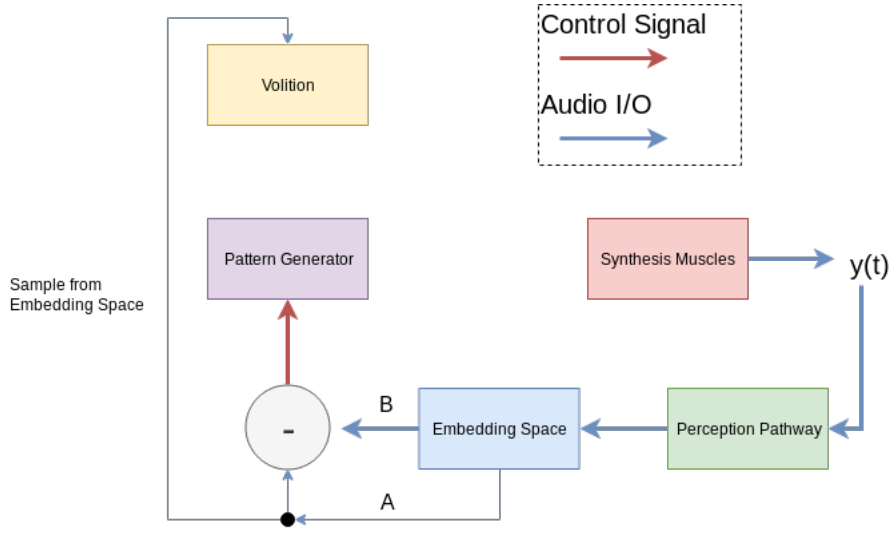

Figure 5.4: ESC Production Reference Model: Learning to babble, continued. This figure is identical to Figure 4.10.

against the target embedding(s). The error signal is used to modify the pattern generator and to potentially modify the training regimen as determined by the volition block. This error signal should move towards zero, which implies that the sound output from the model moves towards the sound that was used to create the target embedding(s). The target embeddings will likely only have high confidence for shorter time scales at this point in the model. Sampling from these time scale embeddings and training on them should result in output sounds that are similar to what infants produce at the age of six months.

Two fitness functions were used to test this portion of the model. First, to provide a baseline performance, the encoder was not used, and instead the genetic algorithm was trained to produce sounds that matched target sounds based on a cross correlation procedure. Specifically, both the target sound and the produced sound were scaled to the interval $[0.0, 1.0]$ and then a cross correlation was done between them and the maximum of the resulting signal was used as the fitness for the utterance. The second fitness function used the encoders: a point in one of the autoencoders' latent spaces was chosen and then each utterance produced by the genetic algorithm was encoded to get a position in the same latent space. The inverse

Table 5.2: Ranges of values allowed for each articulator in the Synthesis trials.

Articulator	**Minimum**	**Maximum**	**Articulator**	**Minimum**	**Maximum**
Interarytenoid	0.5	0.5	Hyoglossus	0.0	0.0
Cricothyroid	-1.0	1.0	Styloglossus	0.0	0.5
Vocalis	-1.0	1.0	Genioglossus	0.0	0.0
Thyroarytenoid	-1.0	1.0	UpperTongue	-1.0	0.0
PosteriorCricoarytenoid	0.0	0.0	LowerTongue	-1.0	0.0
LateralCricoarytenoid	0.0	0.0	TransverseTongue	0.0	0.0
Stylohyoid	0.0	0.0	VerticalTongue	0.0	0.0
Thyropharyngeus	0.0	0.0	Risorius	0.0	0.0
LowerConstrictor	0.0	0.0	OrbicularisOris	0.5	1.0
MiddleConstrictor	0.0	0.0	LevatorPalatini	-1.0	1.0
UpperConstrictor	0.0	0.0	TensorPalatini	-1.0	1.0
Sphincter	0.0	0.0	Masseter	-0.5	0.0
Mylohyoid	-1.0	1.0	LateralPtrygoid	0.0	0.0
Buccinator	0.0	0.0			

of the Euclidean distance (i.e., Equation 5.1) was used as the fitness function.

$$F = \frac{1}{\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}} \tag{5.1}$$

Where n is the number of dimensions in the embedding space.

Matrices of shape 28 x 3 result in a very high dimensional space to search. Since many of the articulators have only a small effect on vocalization and articulation, I limited many of them to small or zero ranges. Table 5.2 shows the range each articulator was allowed to vary over for these trials.

Note that the lungs were set to 0.2 at the first time point and then zero at any subsequent time points.

Lastly, in this phase of the model, to make the process of producing sound from a very

large search space easier, the volitional block was not simply a "for loop" like in the cooing phase, but rather it used a sequence of stages. During the cooing phase, laryngeal muscles were learned. After this, their allowed minima and maxima were annealed to within ± 0.05 of the best value found during this phase. During the babbling phase, only the jaw muscles were allowed over their entire normal range, with the laryngeal ones being allowed within their annealed limits. This went on for some number of iterations. After that, the jaw muscles were annealed to have allowed minima and maxima within ± 0.1 of the best value found. Then the nasal group was annealed, then the lingual support group, followed by the lingual main group, and finally the labial group.

Table 5.3 shows which group each articulator is assigned to.

Note that there is some biological justification for this: see [11] for a discussion on how toddlers learn over time to control their jaw and then their lips for speech.

Lastly, after some sufficient amount of training has been done in this phase, another phase is entered, where the model contains a recurrent connection that allows the volitional block to adjust its training regimen based on its own immediate output. The other difference is that the sampling of embedding spaces should be biased towards longer time scales, so that longer utterances are targeted, produced, and trained against. This phase of the model was not implemented as part of this thesis, due to time constraints; it may be a focus for further work.

5.1.2 Software Repository

The implementation of ESC as outlined above is called *ArtieInfant* and is kept in a Git repository on Github at this URL: https://github.com/MaxStrange/ArtieInfant.git. The version used in this thesis can be found in the bibliography here [61].

In the directory structure of the ArtieInfant project, the only folder of interest is the *Artie* directory, which contains the Python source code for this project. Python 3 was used as the language of choice due to it being the de facto programming language of choice for machine learning research. It has libraries that enable faster development time than languages like C

Table 5.3: Group that each articulator belongs to for annealing purposes.

Articulator	**Group**	**Articulator**	**Group**
Interarytenoid	Laryngeal	Hyoglossus	Lingual Support
Cricothyroid	Laryngeal	Styloglossus	Lingual Support
Vocalis	Laryngeal	Genioglossus	Lingual Support
Thyroarytenoid	Laryngeal	UpperTongue	Lingual
PosteriorCricoarytenoid	Laryngeal	LowerTongue	Lingual
LateralCricoarytenoid	Laryngeal	TransverseTongue	Lingual
Stylohyoid	Laryngeal	VerticalTongue	Lingual
Thyropharyngeus	Laryngeal	Risorius	Labial
LowerConstrictor	Laryngeal	OrbicularisOris	Labial
MiddleConstrictor	Laryngeal	LevatorPalatini	Nasal
UpperConstrictor	Laryngeal	TensorPalatini	Nasal
Sphincter	Laryngeal	Masseter	Jaw
Mylohyoid	Lingual Support	LateralPtrygoid	Jaw
Buccinator	Labial		

or C++, even though those languages make for much faster runtime. The system makes use of TravisCI for unittesting before any merges are made to the master Git branch. These tests are in the *Artie/tests* directory, and the .travis.yml file that controls the testing is found in the root directory of the repository. The project has been tested against Python version 3.6, and has been informally tested against Python 3.5 and Python 3.7. For the specific versions of all the libraries used, see the requirements.txt file in the repository's root directory.

I wrote ArtieInfant to operate as a software framework for testing ESC. This means that the code is reusable for most alterations to ESC - such as modifying, removing, or adding a new signal chain element. The general workflow for testing ESC via ArtieInfant is the

following:

1. Make necessary source code changes (usually merely additions).

2. Create a new experiment configuration file in the *experiment* folder.

3. Run Python on main.py, found in the *Artie* folder, while passing in meta arguments (such as logging information).

4. Results are analyzed automatically as part of running main.py.

Most parameters of the model and directory locations are specified in the experimental configuration file.

5.2 Data

This section describes the data that was collected to test the model. One major advantage that ESC has over other theories of language acquisition is its ability to deal with raw, continuous speech as its input data. Thus, raw, continuous speech was recorded for the input data.

In the Summer of 2017, I had a son (Oliver). From the first week through the ninth month of his life, a caregiver recorded the acoustic environment around him using a digital recording device. The first recording was on June 6th of 2017 (he was born on the 3rd) and the last recording was on February 24th of 2018. By the time the recordings ceased, he was engaging in non-reduplicated babbling. The person responsible for recording was in almost all cases myself, his mother, or his grandmother. From the day of the first recording until January of 2018, he lived with me, his mother, his maternal grandfather, his maternal grandmother, his maternal grandfather's mother, and an uncle on his mother's side, where American English and Taiwanese Mandarin were spoken at about a 50/50 ratio. He had regular (generally weekly) visits to his American grandparents' home, where American English is spoken exclusively. At the beginning of January 2018, we moved in with his paternal

Table 5.4: Descriptive statistics on the dataset used in this thesis.

Statistic	**Value**
Total number of recordings	1942
Total duration	913 hours or 38 days
Average length of a recording	28.2 minutes
Maximum length of a recording	4.75 hours
Average amount of silence in a recording	16.5 minutes
Average number of recordings per day	8.16
Standard deviation of recordings per day	4.42

grandparents, so that he was living with me, his mother, his paternal grandfather, his paternal grandmother, and an aunt on his father's side. These recordings were made while Oliver was awake, though there are occasions when the recorder was left on accidentally. These cases generally resulted in long periods of silence that are easily filtered out.

Table 5.4 gives some descriptive statistics about the recordings that were made.

The microphone that was used to record the data was an EVISTR L57. Encoding was done with WAV at 16 bits per sample, with a 48 kHz sampling rate. I transferred the data from the microphone to a hard drive at an interval of about 14 days. Once on the hard drive, the sound files were also backed up to a cloud storage service to secure against accidental loss of data. By the time Oliver was 9 months old, this resulted in 1942 files of raw audio, of range 0.5 seconds to 4.8 hours, with an average of 28.2 minutes per recording. The total amount of audio recorded was 913 hours, or about 38 days of continuous sound.

Due to the enormous amount of data collected, no attempt was made to label the data by hand. As such, the contents of the Oliver dataset can currently only be estimated. Unfortunately, because the contents are not fully known, the dataset cannot be opensourced at this time. Because this dataset is unique in its capturing of the natural acoustic environment of a child in a bilingual environment in the first nine months of life, I may listen to much

of it and redact 1) voices of people who do not give permission to share them and 2) any identifying details.

5.3 Experiments

This section describes the experiments that I have so far undertaken to use the model (ArtieInfant) to test the theory (ESC). The next chapter describes the results of these experiments, and the chapter after that discusses these results.

All experiments were run using Ubuntu 16.04 LTS on an Intel x86_64 i5-6500 3.2 GHz CPU. This computer was equipped with an NVIDIA GeForce GTX 780. The installed CUDA library was version 9.0.176 with CUDNN version 7.4.2.

Tables 5.5, 5.6, 5.7, and 5.8 show the architectures that were used for the convolutional autoencoders for spectrograms of 241x20x1 dimensions and 81x18x1 dimensions respectively.

5.3.1 Assumption: Embedding Spaces I

I tested the assumption that phonemes will not form obviously separated clusters in the autoencoder's embedding space. To test this, I built several different autoencoders and tried varying amounts and slices of the input dataset.

Vanilla Autoencoder

I tested a deep convolutional autoencoder as specified in Table 5.5, Table 5.6, Table 5.7, and Table 5.8. The network architectures were chosen based on tuning alone — neither the theory nor the reference design specifies any particular network architecture. The reconstruction loss was mean squared error, given by Equation 5.2.

$$L_r = \frac{1}{N}\sum_{i}^{N}(\hat{y_i} - y_i)^2 \tag{5.2}$$

Where N is the number of pixels in the image, $\hat{y_i}$ is the autoencoder's prediction for the ith pixel, and y_i is the true value for that pixel.

Table 5.5: The default encoder architecture for 241x20x1 spectrograms (which corresponds to 0.5 seconds per spectrogram).

Layer	**Output Shape**	**Kernel Shape**	**Stride**	**N filters**
Input	(241, 20, 1)	-	-	-
2D Convolution	(117, 19, 128)	8 x 2	2 x 1	128
Batch Normalization	(117, 19, 128)	-	-	-
2D Convolution	(55, 18, 64)	8 x 2	2 x 1	64
Batch Normalization	(55, 18, 64)	-	-	-
2D Convolution	(24, 17, 32)	8 x 2	2 x 1	32
Batch Normalization	(24, 17, 32)	-	-	-
2D Convolution	(8, 8, 32)	9 x 2	2 x 2	32
Batch Normalization	(8, 8, 32)	-	-	-
2D Convolution	(6, 6, 16)	3 x 3	1 x 1	16
Batch Normalization	(6, 6, 16)	-	-	-
2D Convolution	(4, 4, 8)	3 x 3	1 x 1	8
Flatten	128	-	-	-
Dense	128	-	-	-

Variational Autoencoder

Since variational autoencoders, due to their stringent regularization, are more efficient with the use of their embedding spaces, I also tried variational autoencoders. The architecture was the same as the vanilla autoencoder, with the exception that instead of an N-dimensional embedding vector being the output of the encoder, the encoder output was two different vectors: an N-dimensional mean and an N-dimensional log variance vector. These vectors were combined to produce a multivariate Gaussian distribution by means of Equation 5.3.

$$G_i(\mu, \sigma) = \mu_i + \exp(\sqrt{v_i}) \tag{5.3}$$

Table 5.6: The default decoder architecture for 241x20x1 spectrograms (which corresponds to 0.5 seconds per spectrogram).

Layer	**Output Shape**	**Kernel Shape**	**Stride**	**N filters**
Input	Embedding Dimension	-	-	-
Dense	128	-	-	-
Reshape	(4, 4, 8)	-	-	-
UpSampling	(8, 4, 8)	-	2 x 1	-
Batch Normalization	(8, 4, 8)	-	-	-
2D Convolution	(8, 4, 8)	3 x 3	1 x 1	8
UpSampling	(16, 8, 8)	-	2 x 2	-
Batch Normalization	(16, 8, 8)	-	-	-
2D Convolution	(16, 8, 32)	3 x 3	1 x 1	32
UpSampling	(32, 8, 32)	-	2 x 1	-
Batch Normalization	(32, 8, 32)	-	-	-
2D Convolution	(32, 8, 64)	3 x 3	1 x 1	64
UpSampling	(64, 8, 64)	-	2 x 1	-
Batch Normalization	(64, 8, 64)	-	-	-
2D Convolution	(64, 8, 128)	3 x 3	1 x 1	128
UpSampling	(128, 8, 128)	-	2 x 1	-
Batch Normalization	(128, 8, 128)	-	-	-
2D Convolution	(121, 5, 64)	8 x 4	1 x 1	64
Batch Normalization	(121, 5, 64)	-	-	-
UpSampling	(121, 10, 64)	-	1 x 2	-
2D Convolution	(121, 10, 32)	8 x 2	1 x 1	32
Batch Normalization	(121, 10, 32)	-	-	-
UpSampling	(242, 20, 32)	-	2 x 2	-
2D Convolution	(241, 20, 1)	2 x 1	1 x 1	1

Table 5.7: The default encoder architecture architecture for 81x18x1 spectrograms (which corresponds to 0.3 seconds per spectrogram).

Layer	**Output Shape**	**Kernel Shape**	**Stride**	**N filters**
Input	(81, 18, 1)	-	-	-
2D Convolution	(37, 16, 128)	8 x 3	2 x 1	128
Batch Normalization	(37, 16, 128)	-	-	-
2D Convolution	(16, 14, 64)	6 x 3	2 x 1	64
Batch Normalization	(16, 14, 64)	-	-	-
2D Convolution	(11, 12, 64)	6 x 3	1 x 1	64
Batch Normalization	(11, 12, 64)	-	-	-
2D Convolution	(8, 9, 32)	4 x 4	1 x 1	32
Batch Normalization	(8, 9, 32)	-	-	-
2D Convolution	(5, 6, 32)	4 x 4	1 x 1	32
Batch Normalization	(5, 6, 32)	-	-	-
2D Convolution	(3, 4, 16)	3 x 3	1 x 1	16
Flatten	192	-	-	-
Batch Normalization	192	-	-	-
Dense	128	-	-	-

For $i \in N$, where N is the number of dimensions in the multivariate. The decoder's input was a sample drawn from this distribution. While this task is more difficult than that of simple reconstruction from a compressed representation as in the vanilla autoencoder, the VAE may learn to simply shrink the standard deviations to nothing, while separating the means of the input by significant values. Traditionally (and in this experiment), the VAE's loss function is the average of the reconstructive loss (in this case, mean squared error) and a Kullback-Leibler divergence error term, given by Equation 5.4.

Table 5.8: The default decoder architecture for 81x18x1 spectrograms (which corresponds to 0.3 seconds per spectrogram).

Layer	**Output Shape**	**Kernel Shape**	**Stride**	**N filters**
Input	Embedding Dimension	-	-	-
Dense	128	-	-	-
Reshape	(4, 4, 8)	-	-	-
UpSampling	(8, 4, 8)	-	2 x 1	-
Batch Normalization	(8, 4, 8)	-	-	-
2D Convolution	(8, 4, 8)	3 x 3	1 x 1	8
UpSampling	(16, 8, 8)	-	2 x 2	-
Batch Normalization	(16, 8, 8)	-	-	-
2D Convolution	(11, 6, 32)	6 x 3	1 x 1	32
UpSampling	(22, 12, 32)	-	2 x 2	-
Batch Normalization	(22, 12, 32)	-	-	-
2D Convolution	(9, 10, 32)	6 x 3	2 x 1	32
UpSampling	(18, 20, 32)	-	2 x 2	-
Batch Normalization	(18, 29, 32)	-	-	-
2D Convolution	(7, 18, 64)	6 x 3	2 x 1	64
UpSampling	(14, 18, 64)	-	2 x 1	-
Batch Normalization	(14, 18, 64)	-	-	-
2D Convolution	(14, 18, 64)	8 x 3	1 x 1	64
UpSampling	(28, 18, 64)	-	2 x 1	-
Batch Normalization	(28, 18, 64)	-	-	-
2D Convolution	(28, 18, 64)	8 x 3	1 x 1	64
UpSampling	(56, 18, 64)	-	2 x 1	-
Batch Normalization	(56, 18, 64)	-	-	-
2D Convolution	(49, 18, 64)	8 x 1	1 x 1	64
UpSampling	(98, 18, 64)	-	2 x 1	-
Batch Normalization	(98, 18, 64)	-	-	-
2D Convolution	(91, 18, 64)	8 x 1	1 x 1	64
Batch Normalization	(91, 18, 64)	-	-	-
2D Convolution	(84, 18, 32)	8 x 1	1 x 1	32
Batch Normalization	(84, 18, 32)	-	-	-
2D Convolution	(82, 18, 16)	3 x 1	1 x 1	16
Batch Normalization	(82, 18, 16)	-	-	-
2D Convolution	(81, 18, 1)	2 x 1	1 x 1	1

$$L_{kl} = -\frac{1}{2}\sum_{i}^{N}(1 + v_i - \mu_i^2 - \exp(v_i)) \tag{5.4}$$

Where N is the number of dimensions in the multivariate Gaussian, μ_i is the mean of

the Gaussian in the ith dimension, and v_i is the log variance in the ith dimension. Thus, the loss function was given by Equation 5.5.

$$L_{vae} = \frac{1}{2}(L_{kl} + L_r) \tag{5.5}$$

Overfitting

To see what effect the amount of training had on the embeddings, I tried overfitting and underfitting the autoencoder. To overfit the autoencoder, the data was fed in and its loss score was monitored. After each epoch, the model was validated against a hold-out validation split of the dataset. Overfitting was defined as the point at which the training loss and the validation loss diverged (the training loss kept converging on zero, while the validation loss started increasing away from zero). See Table 5.9 for the complete list of experiments done to test the effects of overfitting on the learned embedding space.

In this table, and the following ones, the *type* of the autoencoder is either *VAE* for variational autoencoder, or *AE* for the vanilla variant. *Dim* is the number of embedding dimensions - in many of these experiments, a small number of dimensions was used for visualization purposes. *Loss* is the loss function: *MSE* stands for mean squared error and *KL* stands for the KL divergence term. *N Data* is the number of data points (spectrograms) used per epoch for training. *N Epochs* is the number of epochs (complete revolutions through the *N Data* data points). Lastly, *Spectrogram* is the dimensionality of the input spectrograms.

Underfitting

To test underfitting, various combinations of epochs and data points were tried. See Table 5.10 for the complete listing.

Table 5.9: Experiments done to test the effects of overfitting of the autoencoder on the embedding space.

Type	**Dim**	**Loss**	**N Data**	**N Epochs**	**Spectrogram**
VAE	2	0.5 * (MSE+KL)	10	5,000	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	1,000	1,000	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	10	5,000	81 x 18 x 1
VAE	2	0.5 * (MSE+KL)	1,000	1,000	81 x 18 x 1
AE	2	MSE	10	5,000	241 x 20 x 1
AE	2	MSE	1,000	1,000	241 x 20 x 1
AE	2	MSE	10	5,000	81 x 18 x 1
AE	2	MSE	1,000	1,000	81 x 18 x 1

Table 5.10: Experiments done to test the effects of overfitting of the autoencoder on the embedding space.

Type	**Dim**	**Loss**	**N Data**	**N Epochs**	**Spectrogram**
VAE	2	0.5 * (MSE+KL)	1,000	1	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	1,000	2	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	10,000	1	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	1,000	1	81 x 18 x 1
VAE	2	0.5 * (MSE+KL)	1,000	2	81 x 18 x 1
VAE	2	0.5 * (MSE+KL)	10,000	1	81 x 18 x 1
AE	2	MSE	1,000	1	241 x 20 x 1
AE	2	MSE	1,000	2	241 x 20 x 1
AE	2	MSE	10,000	1	241 x 20 x 1
AE	2	MSE	1,000	1	81 x 18 x 1
AE	2	MSE	1,000	2	81 x 18 x 1
AE	2	MSE	10,000	1	81 x 18 x 1

Table 5.11: Experiments done to test the effects of loss function on the embedding space.

Type	**Dim**	**Loss**	**N Data**	**N Epochs**	**Spectrogram**
VAE	2	0.9MSE+0.1KL	10,000	50	241 x 20 x 1
VAE	2	0.95MSE+0.05KL	10,000	50	241 x 20 x 1
VAE	2	0.5 * (MSE+STD)	10,000	50	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	10,000	50	241 x 20 x 1
VAE	2	0.9MSE+0.1KL	10,000	50	81 x 18 x 1
VAE	2	0.95MSE+0.05KL	10,000	50	81 x 18 x 1
VAE	2	0.5 * (MSE+STD)	10,000	50	81 x 18 x 1
VAE	2	0.5 * (MSE+KL)	10,000	50	81 x 18 x 1

Loss Function

I hypothesized that the VAE may be able to learn different clusters if it is not as stringently constrained to a normal distribution. Thus, I assigned varying weights to the KL loss term, and in one case even removed it and replaced it with the standard deviation instead (thereby attempting to induce point distributions by having the VAE minimize the standard deviations that it put out). See Table 5.11 for the complete listing of experiments run to test this.

Embedding Dimensions

To determine how the number of embedding dimensions affected the clustering of values in the embedding space, the number of dimensions were varied. See Table 5.12.

5.3.2 Assumption: Embedding Spaces II

I tested the assumption that an autoencoder can be used to form an embedding space that contains most of the information of the spectrograms. See the previous section (Assumption: Embedding Spaces I) and the experiments that were run for it.

Table 5.12: Experiments done to test the effects of varying the number of embedding dimensions.

Type	**Dim**	**Loss**	**N Data**	**N Epochs**	**Spectrogram**
VAE	1	0.5 * (MSE+KL)	10,000	50	241 x 20 x 1
VAE	2	0.5 * (MSE+KL)	10,000	50	241 x 20 x 1
VAE	3	0.5 * (MSE+KL)	10,000	50	241 x 20 x 1
VAE	32	0.5 * (MSE+KL)	10,000	50	241 x 20 x 1
VAE	1	0.5 * (MSE+KL)	10,000	50	81 x 18 x 1
VAE	2	0.5 * (MSE+KL)	10,000	50	81 x 18 x 1
VAE	3	0.5 * (MSE+KL)	10,000	50	81 x 18 x 1
VAE	32	0.5 * (MSE+KL)	10,000	50	81 x 18 x 1
AE	1	MSE	10,000	50	241 x 20 x 1
AE	2	MSE	10,000	50	241 x 20 x 1
AE	3	MSE	10,000	50	241 x 20 x 1
AE	32	MSE	10,000	50	241 x 20 x 1
AE	1	MSE	10,000	50	81 x 18 x 1
AE	2	MSE	10,000	50	81 x 18 x 1
AE	3	MSE	10,000	50	81 x 18 x 1
AE	32	MSE	10,000	50	81 x 18 x 1

5.3.3 Assumption: Synthesis I

I tested the assumption that an articulatory synthesizer can be made to produce sound that changes noticeably over time. To test this assumption, I used the Praat articulatory synthesis model, as previously described. Tables 5.13 and 5.14 show all the experiments that were run to test whether or not this combination of genetic algorithm and articulatory synthesizer could be made to produce noticeably different sounds over time.

In addition to the values in the table, the following values were used in all trials:

1. Percent Selected per Generation: 50.0

2. Mutation: The same mutation function was applied in all cases, and the amount of the genepool that was mutated at each generation was set to 10.0%. The mutation procedure was to replace each value in the given control matrix with a value drawn from a Gaussian distribution described by the starting value as its mean and a standard deviation of 0.1.

In Tables 5.13 and 5.14, *Duration (s)* is the total length of each utterance produced in seconds, *Timepoints* are the points in time (in seconds) when the articulators are activated, *Population* is the number of control matrices in each generation, *N Generations* is the number of generations, *Fitness* is the fitness function that was used (either RMS for root mean square, or XCORR for cross correlation), and *Crossover* is the crossover function that was used. The crossover function bears some additional explanation. In the experiments that follow, either no crossover function was used, or else a 2-point crossover function was used. A 2-point crossover function operates as follows: take two vectors (control matrices are converted to vectors for the genetic algorithm) and choose two points at random on one of the vectors. All items up to the first point are swapped between the two vectors. Then all items after the second point are swapped between the two vectors. This mimics how chromosomes behave during the biological process of meiosis and serves to introduce genetic variability in offspring.

5.3.4 Assumption: Synthesis II

I tested the assumption that the synthesized sounds can be made to change over time to become more similar to a target sound by using the embedding space locations of the sounds as the metric of comparison.

Table 5.13: Parameters used in experiments for testing the assumption that a genetic algorithm and an articulatory synthesis model can be combined to produce noticeably different sounds over time (part I).

Duration (s)	**Timepoints**	**Population**	**N Generations**	**Fitness**	**Crossover**
0.5	0.0	100	12	RMS	None
0.5	0.0, 0.2, 0.4	100	4	RMS	None
0.5	0.0, 0.2, 0.4	100	50	RMS	None
0.5	0.0, 0.2, 0.4	100	20	RMS	2-Point
0.5	0.0	100	12	XCORR	None
0.5	0.0, 0.2, 0.4	100	4	XCORR	None
0.5	0.0, 0.2, 0.4	100	25	XCORR	2-point
0.5	0.0, 0.2, 0.4	100	50	XCORR	2-Point
0.5	0.0, 0.2, 0.4	300	100	XCORR	2-Point

Table 5.14: Parameters used in experiments for testing the assumption that a genetic algorithm and an articulatory synthesis model can be combined to produce noticeably different sounds over time (part II).

Duration (s)	**Timepoints**	**Population**	**N Generations**	**Fitness**	**Crossover**
0.3	0.0	100	12	RMS	None
0.3	0.0, 0.1, 0.25	100	4	RMS	None
0.3	0.0, 0.1, 0.25	100	12	RMS	None
0.3	0.0, 0.1, 0.25	100	50	RMS	None
0.3	0.0, 0.1, 0.25	100	12	RMS	2-Point
0.3	0.0	100	12	XCORR	None
0.3	0.0, 0.1, 0.25	100	4	XCORR	None
0.3	0.0, 0.1, 0.25	100	12	XCORR	None
0.3	0.0, 0.1, 0.25	100	50	XCORR	None
0.3	0.0, 0.1, 0.25	100	12	XCORR	2-Point

Table 5.15: Parameters used in experiments for testing whether the Euclidean distance fitness function could be used to produce sounds that become more similar to target sounds over time.

Duration (s)	**Timepoints**	**Population**	**N Generations**
0.5	0.0	100	12
0.5	0.0, 0.2, 0.4	100	12
0.5	0.0, 0.2, 0.4	300	100
0.3	0.0	100	12
0.3	0.0, 0.1, 0.25	100	12

5.3.5 Prediction: Cooing

I tested the prediction that the synthesized sounds that are output as part of learning to coo will have similar phonetic properties to cooing. I compared the sounds produced during the RMS fitness function genetic algorithm against cooing sounds from the dataset. This analysis is qualitative and is discussed in the following chapters.

5.3.6 Prediction: Marginal Babbling and Reduplicated Babbling

This prediction states that the sounds that the articulatory synthesis model will make as part of learning to babble will have similar phonetic properties to first marginal babbling, and then to reduplicated babbling. While the recurrent neural network was not implemented as part of this work, and therefore the reduplicated babbling portion of this prediction could not be tested, the prediction that this phase leads to marginal babbling was tested in a qualitative manner similar to the cooing prediction mentioned above.

5.3.7 Prediction: Non-Reduplicated Babbling

This prediction states that the sounds that the articulatory synthesis model will make as part of the final phase will have similar phonetic properties to non-reduplicated babbling.

This prediction was not tested as part of this work.

Chapter 6

RESULTS

The previous chapter laid out several experiments that I ran using a software framework for testing the ESC theory of phonology acquisition. This chapter describes the results of these experiments, while the implications of these results are discussed in the next chapter. This chapter is organized as follows: each section describes the results of the experiments that were run to test a particular assumption or prediction.

6.1 Assumption: Embedding Spaces I

This assumption states that clusters of phonemes do not easily form into obvious groups in an embedding space, even with large dimensional embedding spaces and with feature spaces learned from deep neural networks.

Figures 6.1 and 6.2 show the results of training a vanilla autoencoder for 0.5 second utterances, while Figures 6.3 and 6.4 show the results for 0.3 second utterances. These autoencoders were trained on 10,000 spectrograms from the dataset, with 50 epochs each and the spectrograms shown in the figures are from a hold-out test split. These figures show the utterances as spectrograms before entering the appropriate autoencoder on the left, and on the right, they show the spectrograms output by the autoencoder. While not perfect, it can be seen from inspection that the autoencoder was able, at both time lengths, to successfully learn an embedding that retains at least some of the information. Note that the input dimensionality for these spectrograms was $241 * 20 * 1 = 4820$ and $81 * 18 * 1 = 1458$, and that the dimensionality for the embedding spaces in these autoencoders was only 3D.

Figures 6.5 and 6.6 show a visualization of the embedding spaces of these autoencoders, where the blue dots are the spectrograms corresponding to all the utterances in the test

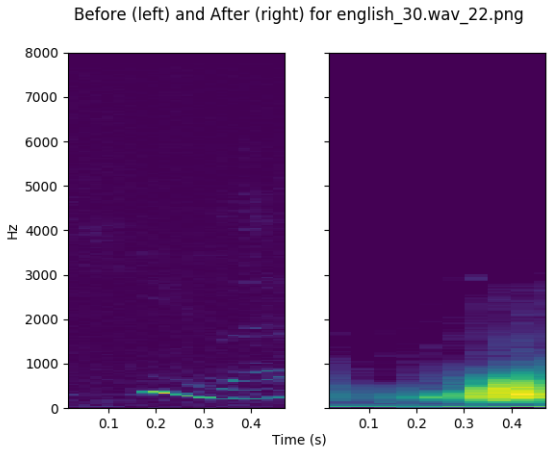

Figure 6.1: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the output of the autoencoder is on the right. Part one of two.

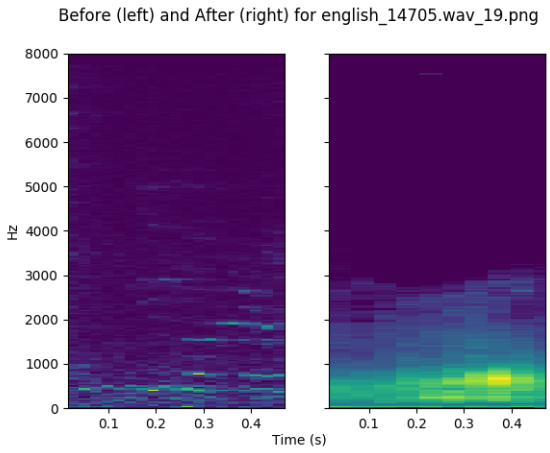

Figure 6.2: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the output of the autoencoder is on the right. Part two of two.

split, of which there were $N = 9976$ for the 241x20x1 spectrograms and $N = 991$ for the 81x18x1 spectrograms. It is interesting to note that they seem to cluster close to the origin. A natural question is whether there is any obvious rhyme or reason to the embeddings. If we

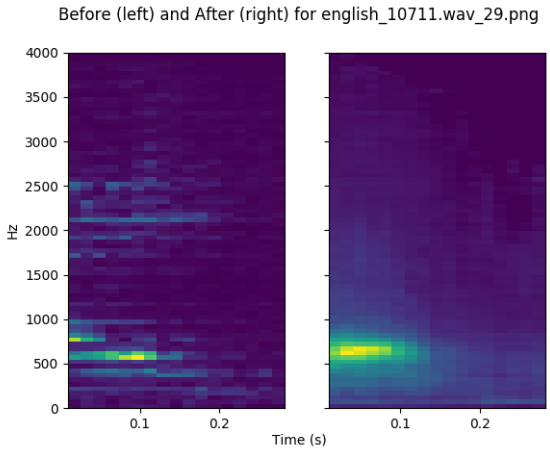

Figure 6.3: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the output of the autoencoder is on the right. Part one of two.

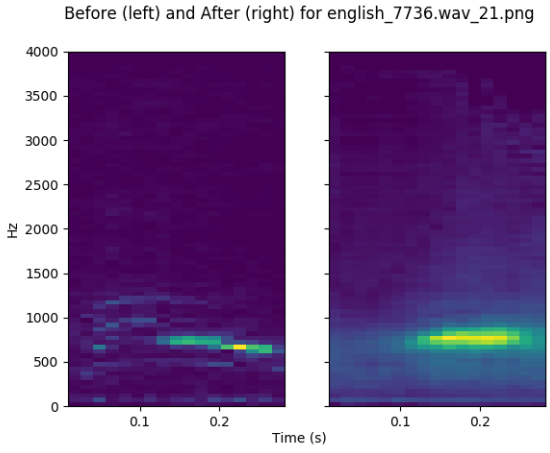

Figure 6.4: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the output of the autoencoder is on the right. Part two of two.

can analyze what types of sounds end up in different locations in the embedding space, we can make some inferences about the topography of the space - do we see any patterns that might help deduce whether minimizing distances between embeddings in this space might

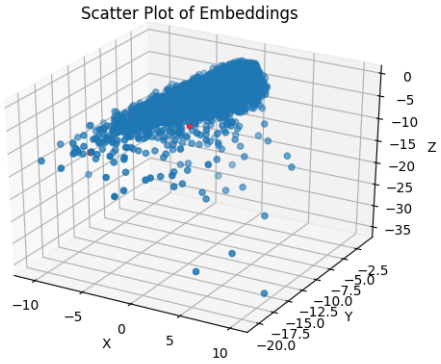

Figure 6.5: Embeddings of test split in embedding space of 241x20x1 spectrogram with English vowels shown in red.

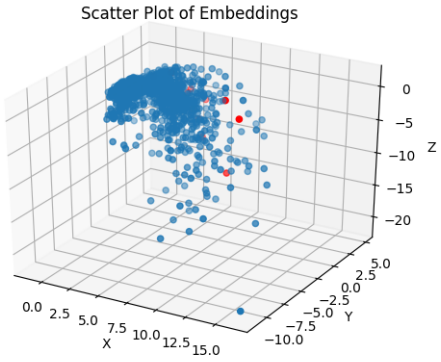

Figure 6.6: Embeddings of test split in embedding space of 81x18x1 spectrogram with English vowels shown in red.

provide a smooth fitness function that could be used as the self-supervising signal in infants? And secondly, can we draw any conclusions about what the basis features are? To start this analysis, I plotted all English vowels, as taken from each vowel's page on Wikipedia. These are shown in red. Several of the vowels are hidden within the point clouds. This seems to show that the vowels do not appear to be anomolous — they are encoded by the

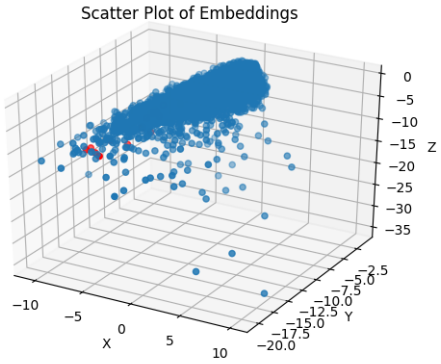

Figure 6.7: Embeddings of a test split in embedding space of 241x20x1 spectrogram, with variations of /a/ highlighted in red.

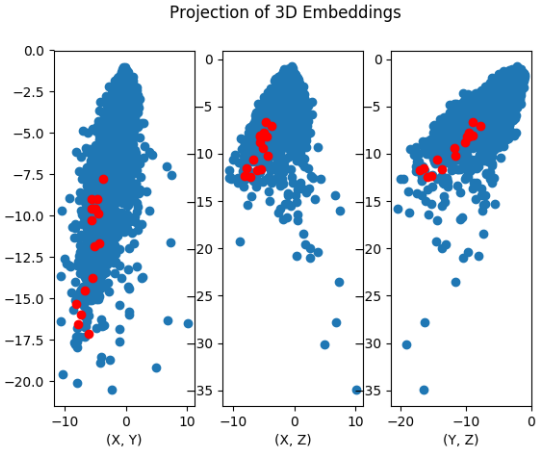

Figure 6.8: Embeddings of a test split in embedding space of 241x20x1 spectrogram, projected on to each plane, with variations of /a/ highlighted in red.

autoencoders right along with all the other items.

Figures 6.7 and 6.9 again show this visualization, but this time the red dots are all the same vowel as perceived by an adult English speaker: specifically they are just the vowel /a/ recorded from myself saying "aah" 16 times into a microphone. Since it is somewhat difficult to tell whether these dots have formed into clusters in these visualizations, I have

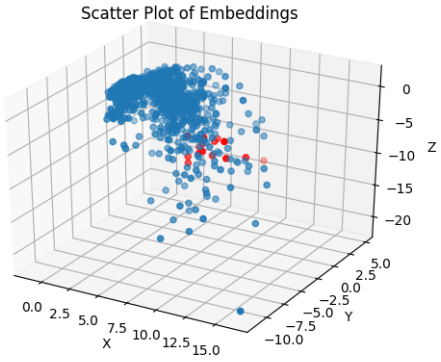

Figure 6.9: Embeddings of a test split in embedding space of 81x18x1 spectrogram, with variations of /a/ highlighted in red.

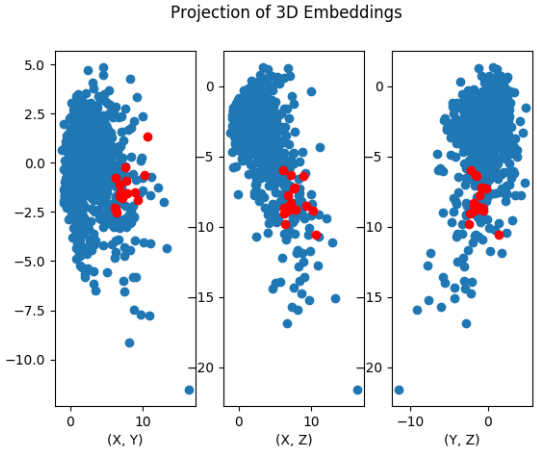

Figure 6.10: Embeddings of a test split in embedding space of 81x18x1 spectrogram, projected on to each plane, with variations of /a/ highlighted in red.

also plotted projections of these onto the (X, Y), (X, Z), and (Y, Z) planes. These can be seen in Figures 6.8 and 6.10. It is interesting to note that the /a/ utterances do seem to have formed a cluster in the 81x18x1 condition, but not in the 241x20x1 condition.

Of more interest for the assumption we are testing are Figures 6.11 and 6.12, which show 11 instances of /f/, where each one is parsed from me saying "Spherical friends with awfully

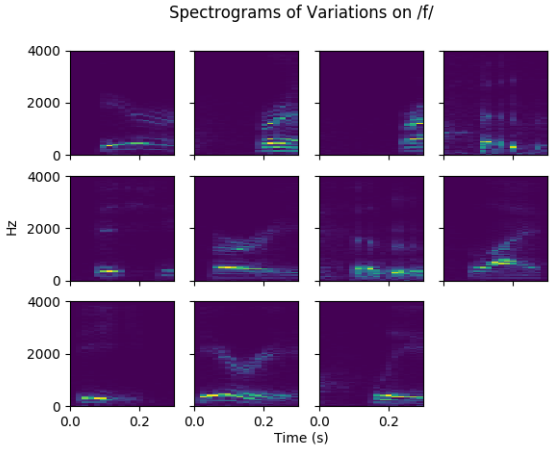

Figure 6.11: Spectrograms of /f/ as parsed from me saying "Spherical friends with awfully furry feet fly furiously while frowning at fishy siphons used quite often".

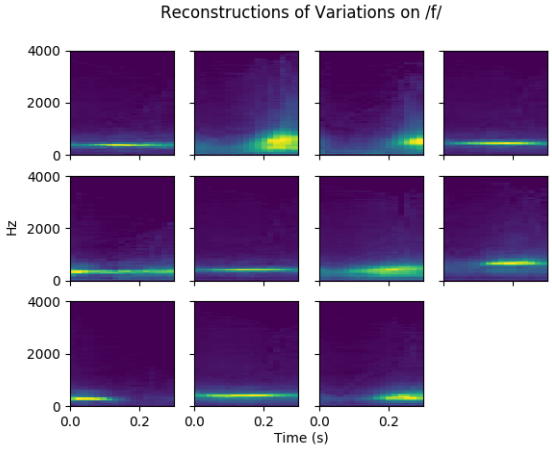

Figure 6.12: Reconstructions by the autoencoder of /f/ as parsed from me saying "Spherical friends with awfully furry feet fly furiously while frowning at fishy siphons used quite often".

furry feet fly furiously while frowning at fishy siphons used quite often". Unfortunately, /f/ is a short phoneme in natural speech, and therefore much of each spectrogram is dominated by whatever sound comes after the /f/ sound. Therefore, it only really makes sense to test the 81x18x1 condition (which had total spectrogram time lengths of 0.3 seconds, as opposed to 0.5 for the 241x20x1 condition).

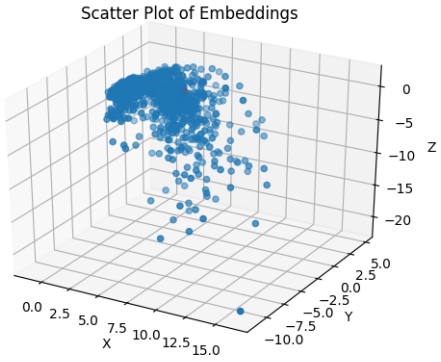

Figure 6.13: Embeddings of test split (blue) and /f/ (red) in 3D embedding space.

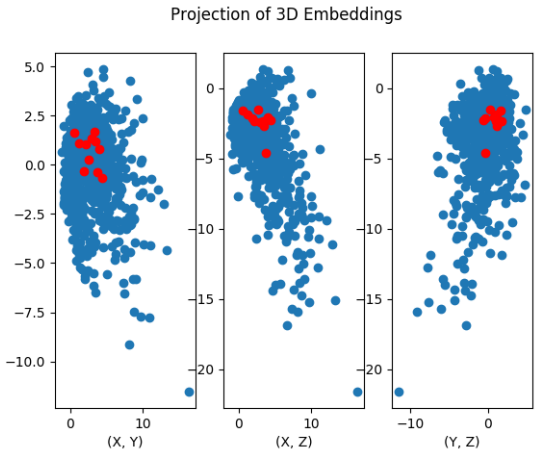

Figure 6.14: Embeddings of test split (blue) and /f/ (red) in 3D embedding space, projected onto each plane.

Figure 6.13 shows the 3D embedding space of the 81x18x1 autoencoder, with the instances of /f/ highlighted in red. Since it is difficult to tell if they form a cluster, Figure 6.14 shows the embedding space projected onto each plane. Intriguingly, these may form a cluster. Since this was unexpected, I further analyzed this result by embedding the 3D embedding space into a 2D t-SNE representation. *T-Distributed Stochastic Neighbor Embedding* or *t-SNE* is

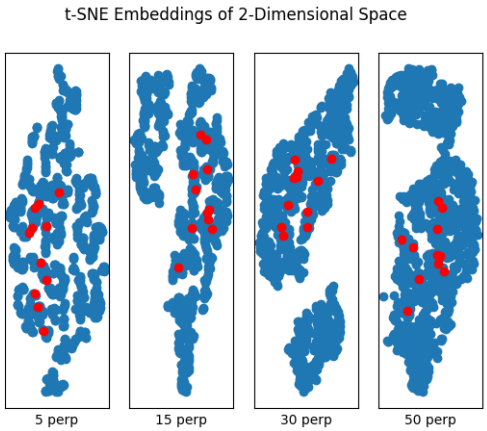

Figure 6.15: Embeddings of test split (blue) and /f/ (red) in 3D embedding space, embedded via t-SNE into 2D at several perplexities.

a method for Visualizing high dimensional data in two or three dimensions. This method maps high dimensional data to a lower dimensional representation while ensuring that a distance function (in our case, the Euclidean distance function) maintains relative values - that is, points that are relatively close to one another in the higher dimensional space will still be close to one another in the low dimensional space, while points that are relatively far from one another in the higher dimensional space will still be relatively far from one another in the lower dimensional space. Since t-SNE is a stochastic method, it is sensitive to hyper parameters. As such, it is good practice to visualize the embedding in more than just one way. In particular, I have adjusted the perplexity value, which adjusts the balance of maintaining the local or global relationships in the data. Low perplexities will typically result in embeddings that are dominated by local relationships, while higher perplexities will typically result in embeddings that are dominated by global relationships.

Figure 6.15 shows the t-SNE embeddings of the 3D embedding space containing /f/ at several perplexity values. This plot seems to indicate that there is no obvious relationship between the /f/ utterances in the 3D embedding space.

Figures 6.16 and 6.17 show the embedding space of a one-dimensional autoencoder, with

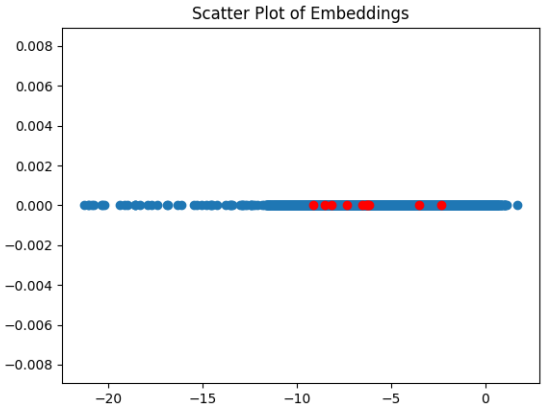

Figure 6.16: Embeddings of test split in embedding space of 241x20x1 spectrogram (1D embedding space), with English vowels shown in red.

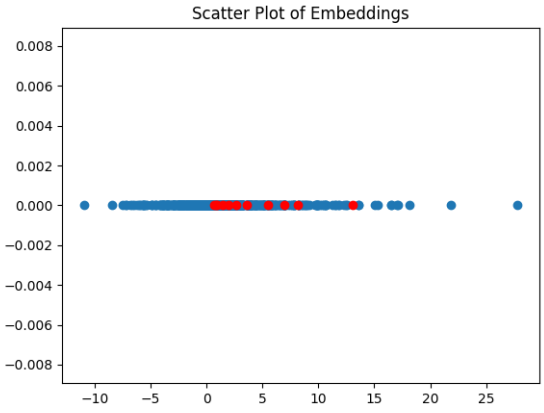

Figure 6.17: Embeddings of test split in embedding space of 81x18x1 spectrogram (1D embedding space), with English vowels shown in red.

red dots corresponding to English vowels.

Compressing spectrograms of 4820 and 1458 dimensions into a single dimension results in severe reconstructive loss (unsurprisingly), as can be seen in Figures 6.18 and 6.19. These figures seem to indicate that the autoencoder simply learned to identify the frequency band of highest power and then decoded that value into a band of high energy across time at the

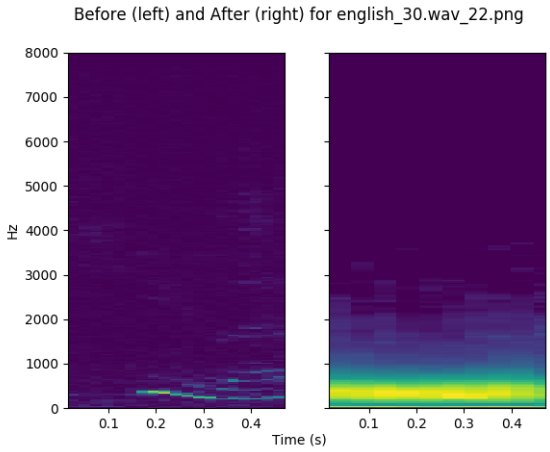

Figure 6.18: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder with a 1D embedding. Input spectrogram is on the left, with the autoencoder's output on the right.

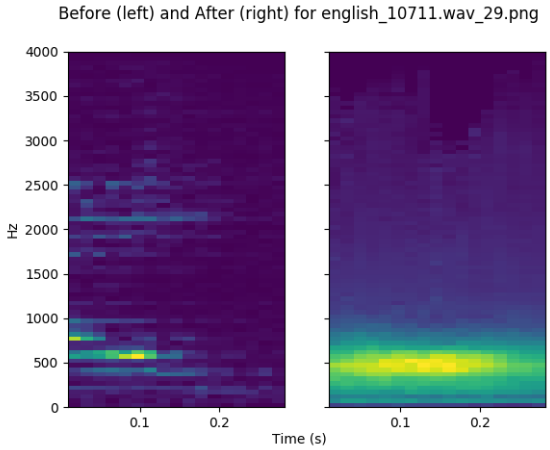

Figure 6.19: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder with a 1D embedding. Input spectrogram is on the left, with the autoencoder's output on the right.

appropriate frequency. This may indicate that the deep convolutional autoencoder, equipped with an MSE loss function, is focusing on the frequency components of most energy.

For completeness, I also investigated the performance and embedding space of 2-dimensional

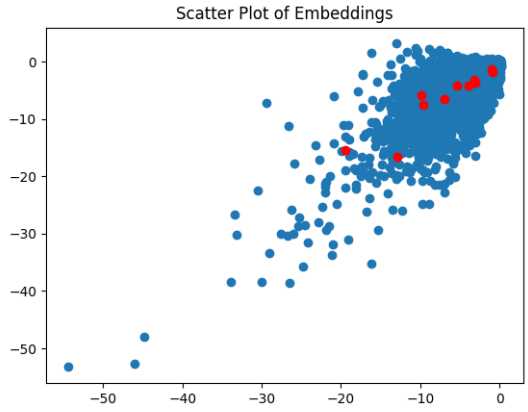

Figure 6.20: Embeddings of test split in embedding space of 241x20x1 spectrogram (2D embedding space), with English vowels shown in red.

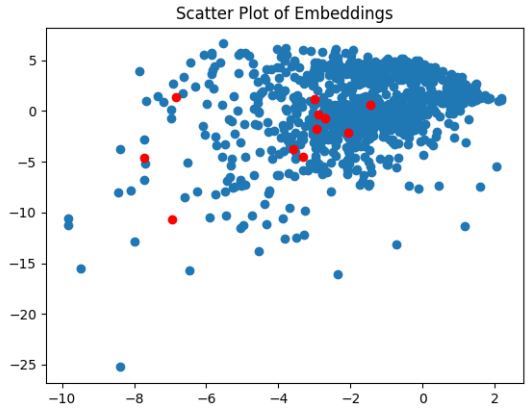

Figure 6.21: Embeddings of test split in embedding space of 81x18x1 spectrogram (2D embedding space), with English vowels shown in red.

autoencoders, which can be seen in Figures 6.20, 6.21, 6.22 and 6.23. Again, the embedding space of these autoencoders is mostly continuous (as opposed to having large distances between embeddings), and the reconstructive ability indicates that some information has been extracted, compressed, and then decompressed.

It feels intuitive that removing dimensions from an embedding space would force the

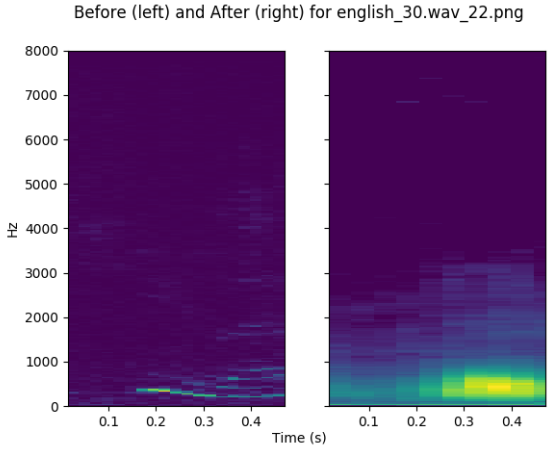

Figure 6.22: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder with a 2D embedding. Input spectrogram is shown on the left, with the autoencoder's output on the right.

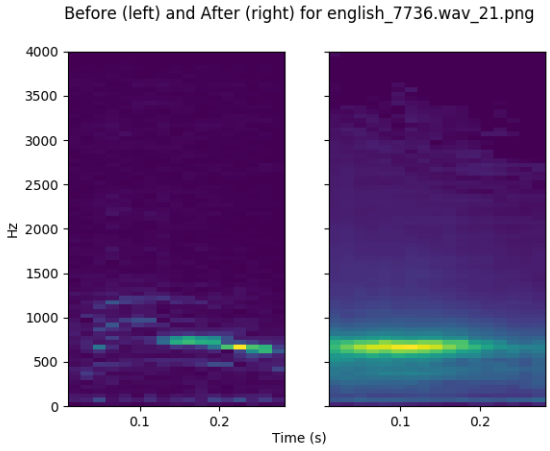

Figure 6.23: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder with a 2D embedding. Input spectrogram is shown on the left, with the autoencoder's output on the right.

basis vectors spanning that space to capture most of the variance in it, if it still operates as a successful embedding space. Indeed, this is the driving idea behind a popular method of dimensionality reduction known as principle component analysis. On the other hand, it may

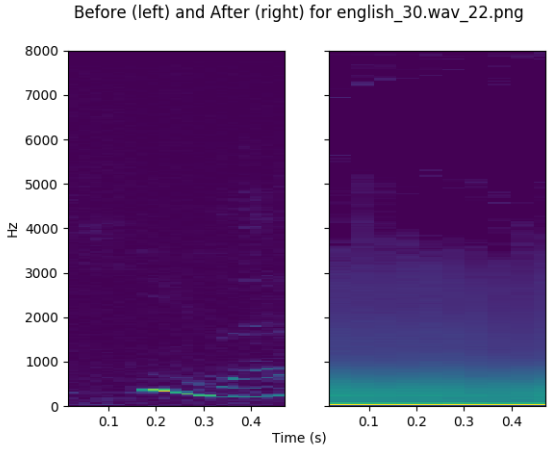

Figure 6.24: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder with a 64D embedding. Input spectrogram is shown on the left, with the autoencoder's output on the right.

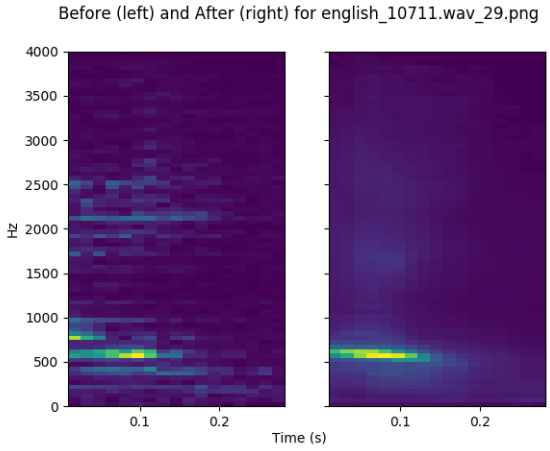

Figure 6.25: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder with a 64D embedding. Input spectrogram is shown on the left, with the autoencoder's output on the right.

be that clusters form after the addition of *more* features to an embedding space. Figures 6.24 and 6.25 show typical examples of reconstruction for 64-dimensional embedding spaces, for the 241x20x1 condition and 81x18x1 condition, respectively. It is interesting that, while the

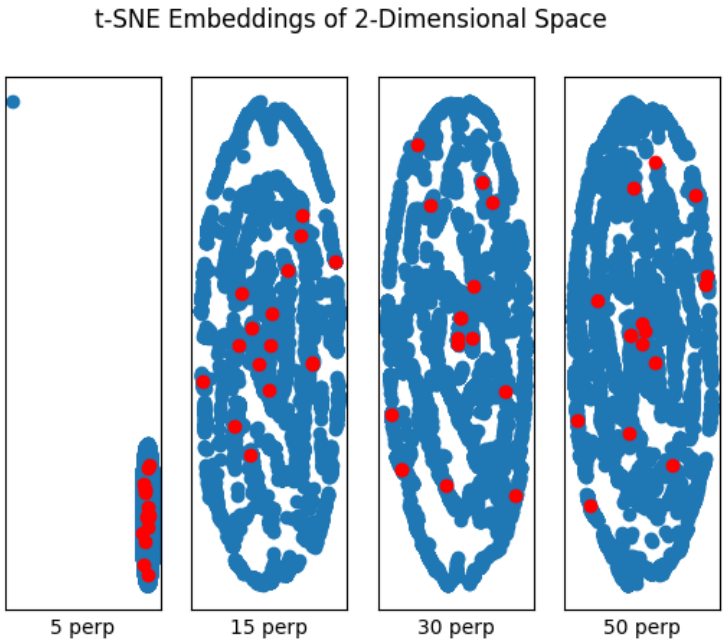

Figure 6.26: Several t-SNE visualizations for 64D embedding space - perplexity values 5, 15, 30, and 50 are given. Red dots indicate relative locations of 16 utterances of the vowel /a/ (241x20x1 condition)

reconstruction is better than in the smaller dimensional spaces, it is still far from perfect, especially in the 241x20x1 condition. This may indicate that the autoencoder's architecture was underpowered. Visualizing a 64-dimensional embedding space does not lend itself to any obvious methods, but some mechanisms do exist for this purpose. T-SNE is one such mechanism, as mentioned previously in this chapter.

Figures 6.26 and 6.27 show the results of t-SNE at various perplexity for the 241x20x1 spectrogram condition and the 81x18x1 spectrogram condition, respectively. The red dots are the 16 /a/ utterances from earlier. As can be seen in the 81x18x1 spectrogram condition, /a/ forms a cluster in the 64-dimensional embedding space. Interestingly, this does not seem to have happened in the 241x20x1 condition. Figure 6.28 shows the same visualization, but for the 81x18x1 condition with /f/ utterances highlighted in red. No clusters seem to have

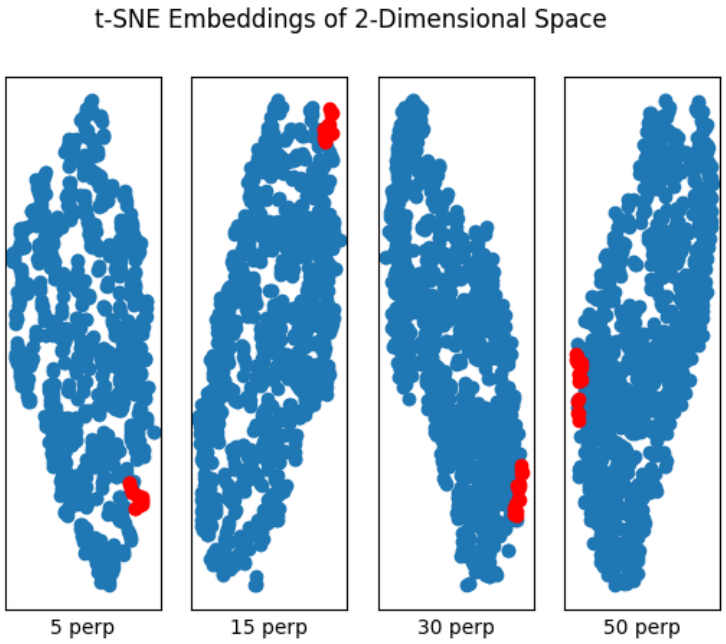

Figure 6.27: Several t-SNE visualizations for 64D embedding space - perplexity values 5, 15, 30, and 50 are given. Red dots indicate relative locations of 16 utterances of the vowel /a/ (81x18x1 condition)

formed in this condition.

Variational autoencoders are a type of autoencoder that have an additional constraint that their embedding spaces conform to a normal distribution, and that their embeddings be merely probabilistic, rather than deterministic. Since this places stringent requirements on the autoencoder's embedding space, it makes sense that the autoencoder would need to learn a very efficient embedding space to be able to reconstruct input spectrograms. Such an efficient embedding space may form clusters due to understanding relationships that are not readily apparent in the input data. Figures 6.29 and 6.30 show results that are analagous to the vanilla autoencoders previously mentioned. Unfortunately, none of the variational autoencoder models performed very well. It is likely that more hyperparameter search is required to achieve good results with these models.

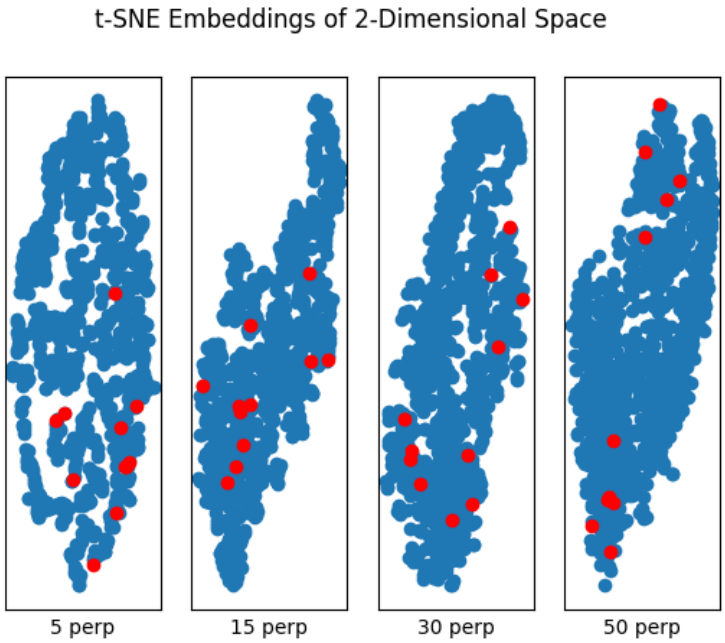

Figure 6.28: Several t-SNE visualizations for 64D embedding space - perplexity values 5, 15, 30, and 50 are given. Red dots indicate relative locations of 11 utterances of /f/ (81x18x1 condition)

6.2 Assumption: Embedding Spaces II

This assumption states that, while phonemes *per se* may not form into clusters in the embedding spaces described, there does exist some algorithm that can learn to embed utterances into a low-dimensional space.

The results from the previous section, especially Figures 6.5 and 6.6 can be used to show that indeed, the deep convolutional autoencoder suffices as an algorithm for learning an embedding space for speech utterances, at least for the time lengths specified for these experiments. I will discuss the reasoning behind this conclusion and its implications in the next chapter.

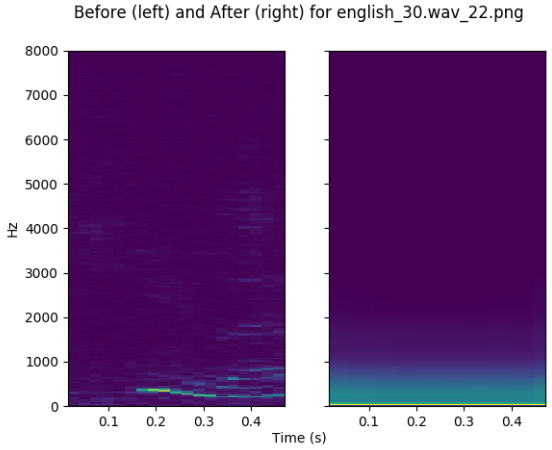

Figure 6.29: Typical spectrogram reconstruction for the 241x20x1 variational convolutional autoencoder. Input spectrogram is on the left, with the autoencoder's output on the right.

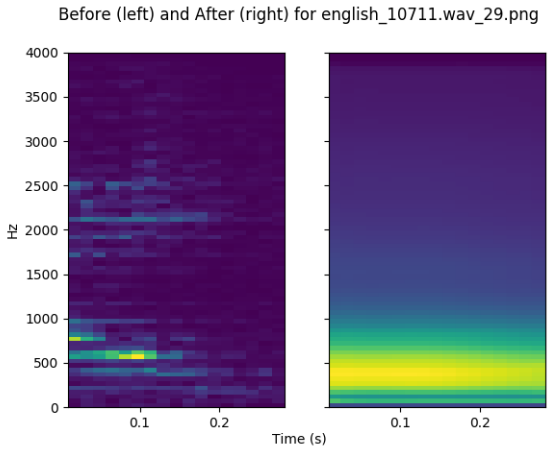

Figure 6.30: Typical spectrogram reconstruction for the 81x18x1 variational convolutional autoencoder. Input spectrogram is on the left, with the autoencoder's output on the right.

6.3 Assumption: Synthesis I

This assumption states that there exists some algorithm that can learn to produce noticeably different speech sounds.

Figure 6.31 shows the results of manually specifying different articulator activations for

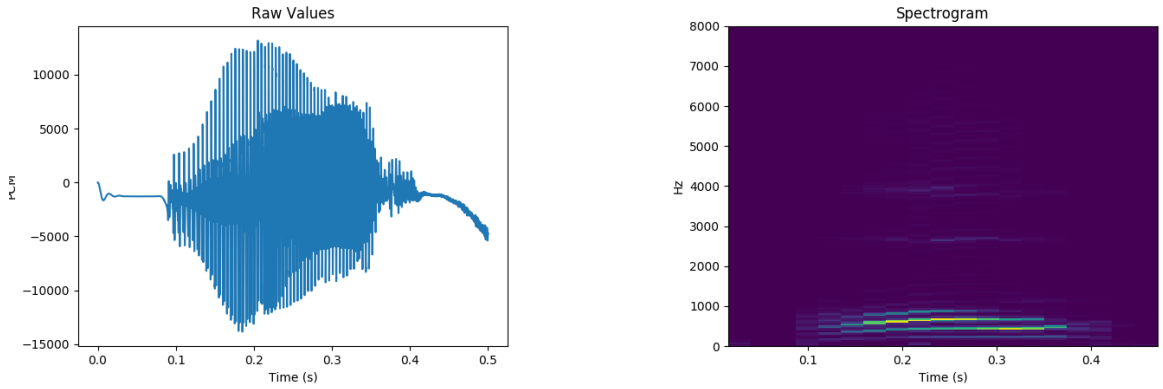

Figure 6.31: Waveform and spectrogram of a manually created utterance with Praat using the same limits as allowed in the genetic algorithm. This utterance sounds something like /o/ ("oh").

the Praat articulatory synthesis model. These utterances were created using the same Python interface (with the same built-in limits on the control matrices) as the genetic algorithm had access to. The point of these plots is to show that demonstrably different sounds are possible with the articulatory synthesis model and interface. However, while these plots show that different noises are attainable via this method, they do not answer whether automating the process of mimicking a target sound with this model is possible.

Using a genetic algorithm to search through the space of possible muscle activations, I first used only the RMS of the resultant sound as a fitness function. Figure 6.32 shows the fitness function's value for the best agent in the gene pool, the average of the whole gene pool, and the worst agent in the gene pool for a single trial: a 50 generation, no crossover trial. It is obvious from the plot that a large enough population of initial agents finds the optimal (or at least, locally optimal) solution simply on random initialization. Due to this, and because training the genetic algorithm using RMS corresponds to the first phase of training in the ESC model (learning to coo), all subsequent genetic algorithm experiments used an RMS pretraining regimen of 16 generations (just to be cautious).

Figure 6.33 shows the spectrogram and the waveform of the best agent from the gene

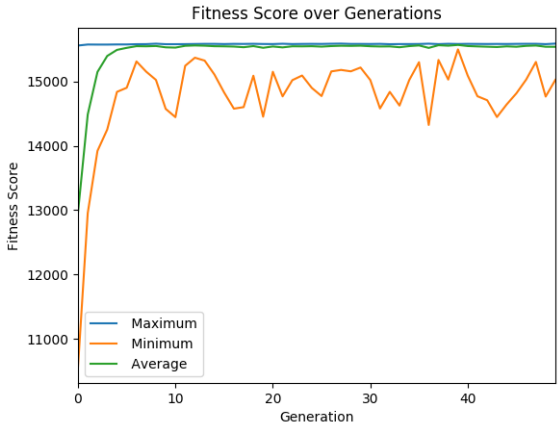

Figure 6.32: Maximum, average, and minimum values for control matrices produced at each population in a genetic algorithm using RMS as the fitness function (50 generations, no crossover).

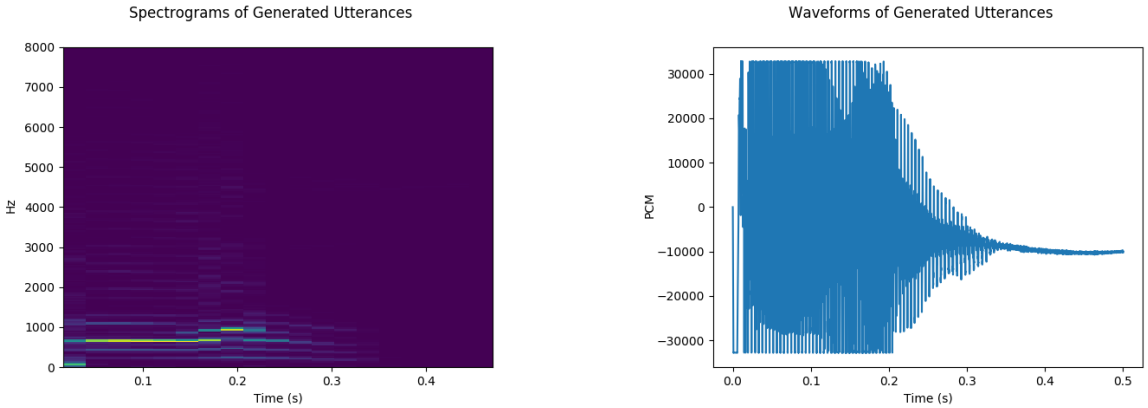

Figure 6.33: Typical spectrogram (left) and waveform (right) of the best agent from a gene pool trained to maximize RMS (0.5 second utterance).

pool at the end of a typical RMS trial. As can be clearly seen from the spectrogram, the model learns to vibrate its vocal chords and output a sound at around 650 Hz, corresponding to /ʌ/, and to do so with large amplitude in the wave form.

Starting from a 16 generation RMS pretraining regimen, several experiments were run in an attempt to mimic target utterances. The experiments were done with 0.5 second utterance

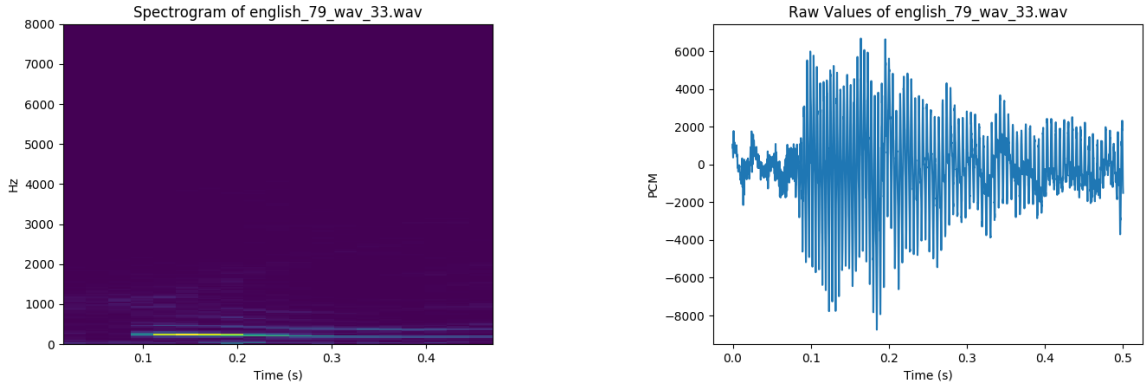

Figure 6.34: Spectrogram (left) and waveform (right) of the first 0.5 second target for the genetic algorithm.

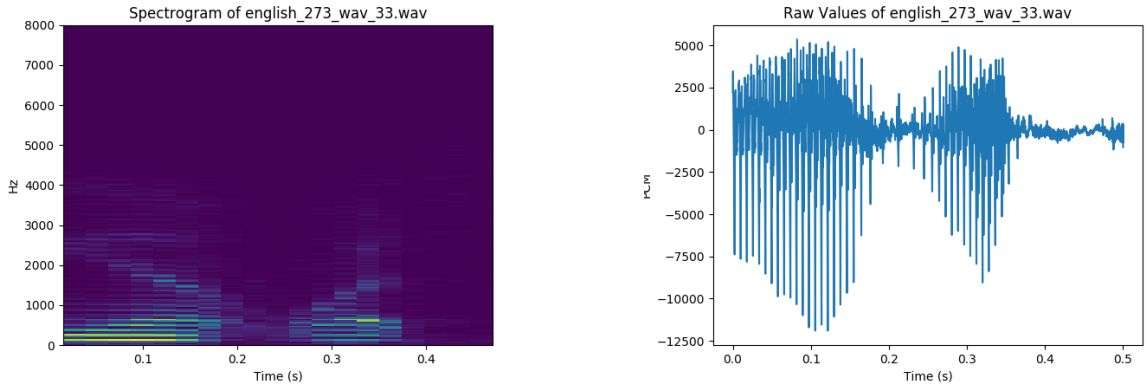

Figure 6.35: Spectrogram (left) and waveform (right) of the second 0.5 second target for the genetic algorithm.

targets and with 0.3 second utterance targets, corresponding to the spectrogram conditions from the autoencoder trials. To try to create artificial speech sounds that sounded different from one another, I used samples of speech from the Oliver dataset as targets to have the genetic algorithm learn to produce. These targets are shown in Figures 6.34, 6.35, 6.36, and 6.37.

Figure 6.34 corresponds to me (an adult male) saying "and watch", though only /ændwa/ is present in the track. This was in a request to my wife to watch television, and so I was

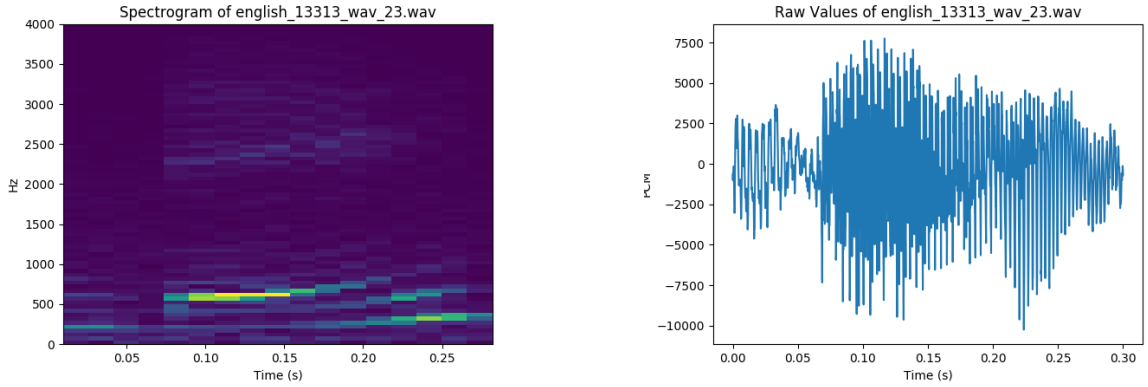

Figure 6.36: Spectrogram (left) and waveform (right) of the first 0.3 second target for the genetic algorithm.

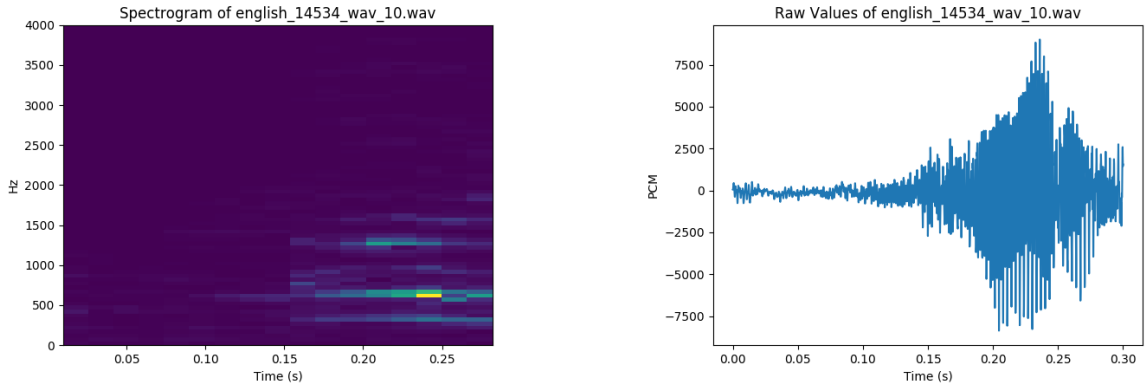

Figure 6.37: Spectrogram (left) and waveform (right) of the second 0.3 second target for the genetic algorithm.

engaging in normal adult-to-adult communication (as opposed to infant directed speech). Figure 6.35 on the other hand, corresponds to me saying "no crying" in infant-directed speech to Oliver (Oliver cannot be heard during the time of this utterance). The track only includes /kɹajiŋ/ ("crying"). Figure 6.36 corresponds to my wife (an adult female) talking to me in normal conversation. The utterance was part of "do you" and captures /du/ ("do"). Lastly, Figure 6.37 corresponds to my mother (an adult female) exclaiming "oh" in response to Oliver spitting up after drinking from a bottle. The track captured /o/ fairly well.

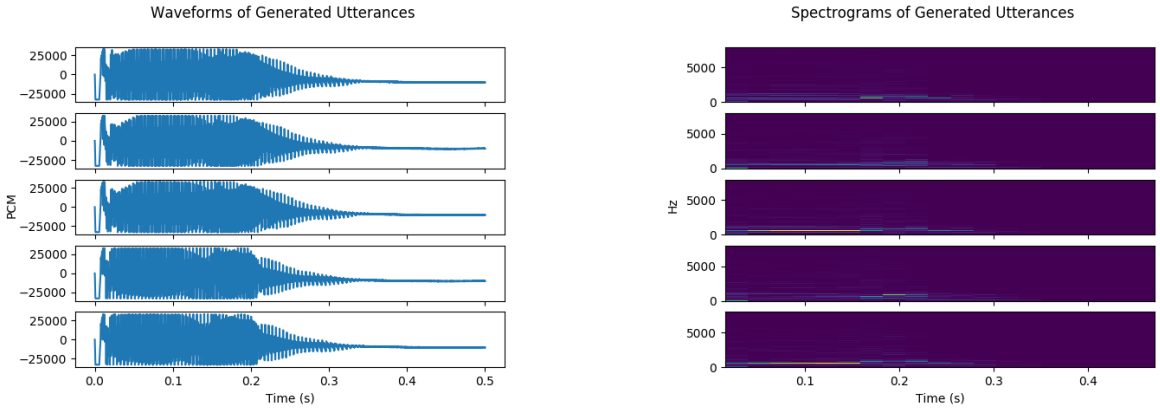

Figure 6.38: Evolution of the waveform and spectrogram of the first 0.5 second target for the genetic algorithm (25 generations, 100 population, cross-correlation fitness function, 2-point crossover).

To determine if the genetic algorithm could learn to mimic target utterances, the normalized cross-correlation method was used (as described in the previous chapter) as a fitness function. The results of this are shown in Figure 6.38 for the first 0.5 second target utterance, at five different times during training. Time goes from top to bottom in these figures, so the topmost plot in a figure is the earliest of the five samples. The evolution of this sound is not obvious from looking at it, though its maximum fitness score does increase over time.

A better example is given by Figure 6.39, which shows the same analysis, but for the second target in the 0.5 second utterance condition. Although the spectrogram is hard to see, the waveform evolution is clear - increasing the cross-correlation can be brought about in this case by silencing the utterance after a short time, corresponding to the quiet period in the middle of the target utterance. It is possible that a second lobe would have appeared later on in the waveform if the algorithm had been allowed to run for longer.

Figure 6.40 again shows the same analysis, but in this case for the first target in the 0.3 second utterance condition. It is not obvious from these images that the genetic algorithm managed to perform better than the RMS pretraining regimen alone did.

Lastly, Figure 6.41 shows the same results for the second target in the 0.3 second utterance

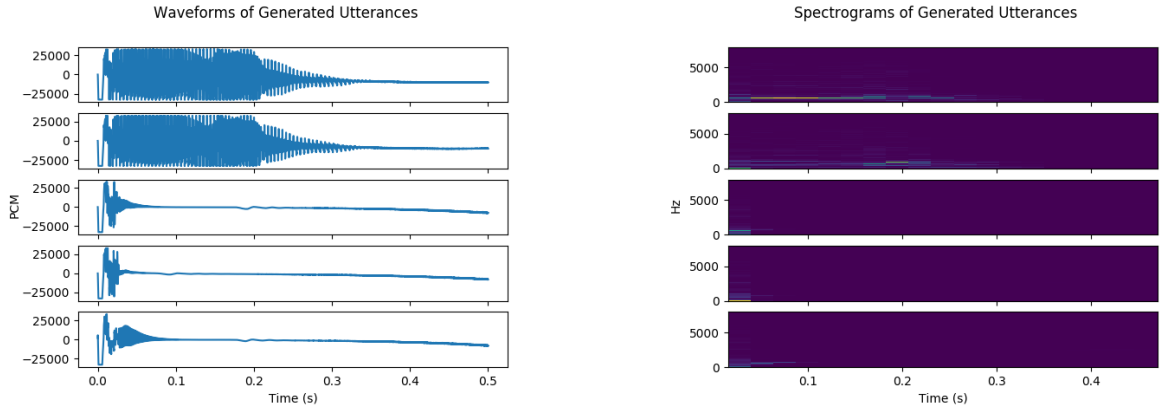

Figure 6.39: Evolution of the waveform and spectrogram of the second 0.5 second target for the genetic algorithm (25 generations, 100 population, cross-correlation fitness function, 2-point crossover).

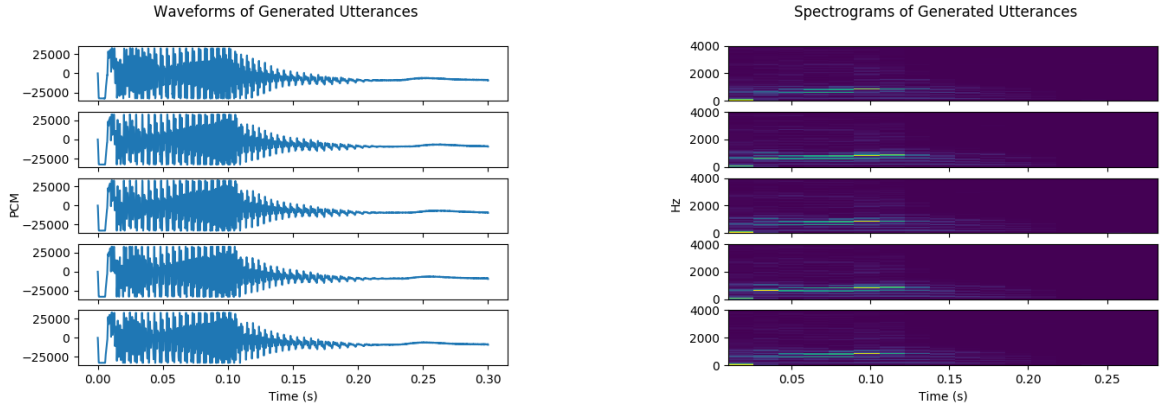

Figure 6.40: Evolution of the waveform and spectrogram of the first 0.3 second target for the genetic algorithm (12 generations, 100 population, cross-correlation fitness function, 2-point crossover).

condition. This target has almost all of its energy in the last half of the 0.3 seconds, which is difficult for the genetic algorithm to reproduce, since it was given only an initial lung activation and was not allowed to manipulate the lungs. Therefore, the algorithm oscillates between two local maxima - producing no sound and producing sound that is elongated.

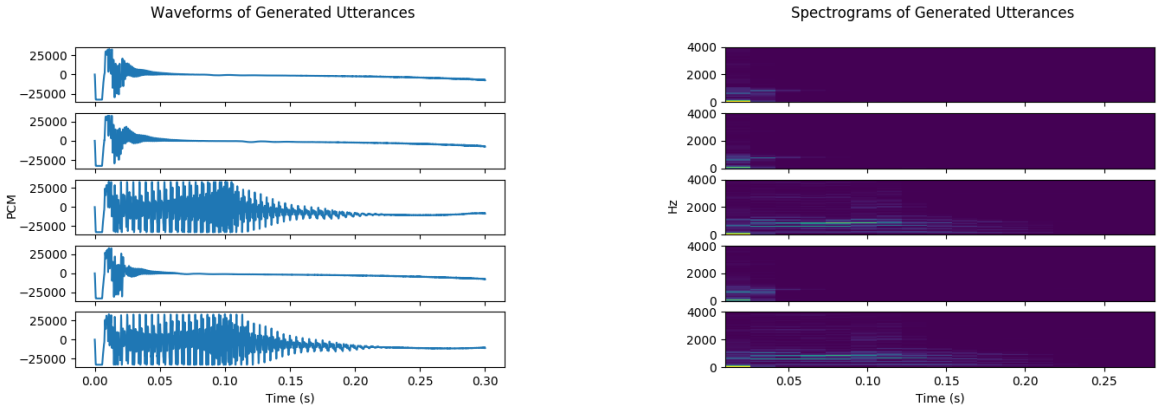

Figure 6.41: Evolution of the waveform and spectrogram of the second 0.3 second target for the genetic algorithm (12 generations, 100 population, cross-correlation fitness function, 2-point crossover).

6.4 Assumption: Synthesis II

This assumption states that the distance between embeddings of a speech sound and an utterance that seeks to resemble that sound can be used as a loss function to train an algorithm to learn to produce the target sound.

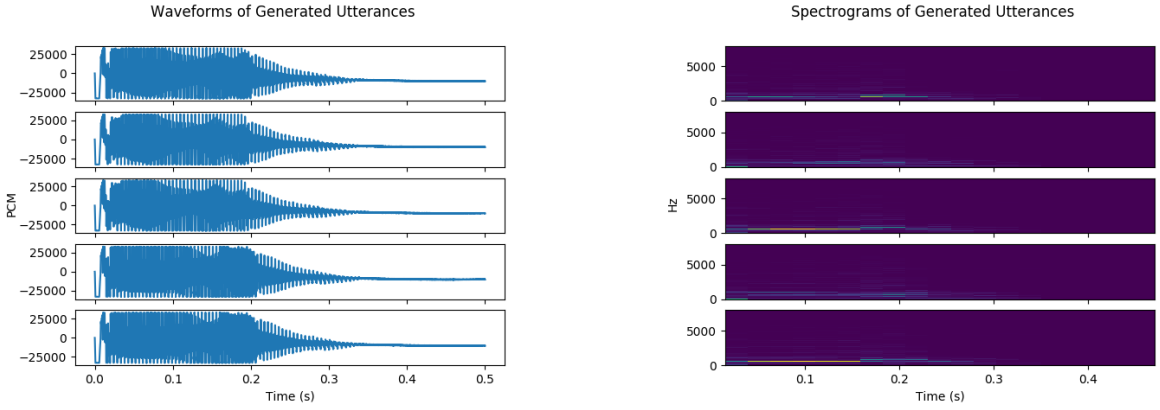

Figure 6.42: Evolution of the waveform and spectrogram of the first 0.5 second target for the genetic algorithm (12 generations, 100 population, Euclidean fitness function, 2-point crossover).

Figure 6.42 shows the same analysis as previously described for the cross-correlation condition, but in this case, the fitness function was the inverse of the Euclidean distance between the generated utterance and the target utterance in a 3-dimensional embedding space determined by the convolutional autoencoder described previously. Comparing this figure to Figure 6.38 is instructive: they are quite similar.

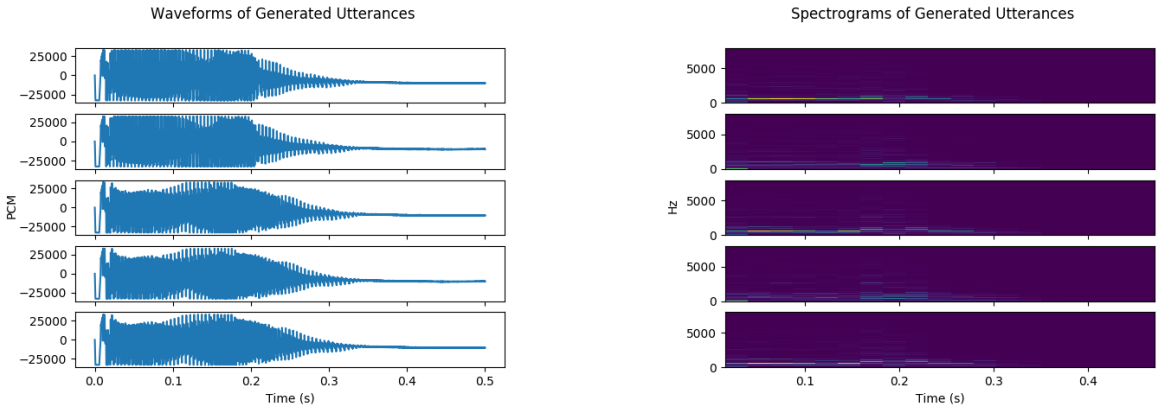

Figure 6.43: Evolution of the waveform and spectrogram of the second 0.5 second target for the genetic algorithm (25 generations, 100 population, Euclidean fitness function, 2-point crossover).

Figure 6.43 again shows the same analysis, but this time for the second target in the 0.5 second condition. This figure diverges drastically from Figure 6.39. However, they also seem to be beginning to form two lobes, like in the target; though the trial did not go on long enough to say conclusively whether this would have happened.

Figure 6.44 shows the same analysis, but for the first target in the 0.3 second condition. This figure seems again to show a similar situation to its analogous plot, Figure 6.40. Again, the RMS pretraining seems to have produced a sound that needs only minor corrections.

Lastly, Figure 6.45 shows the same results as the cross-correlation trials once more, but this time for the second target in the 0.3 second condition. In this case, the genetic algorithm seems to have learned to move the spectral energy in time towards the end of the utterance.

Unfortunately, these experiments are computationally intensive, since at each generation,

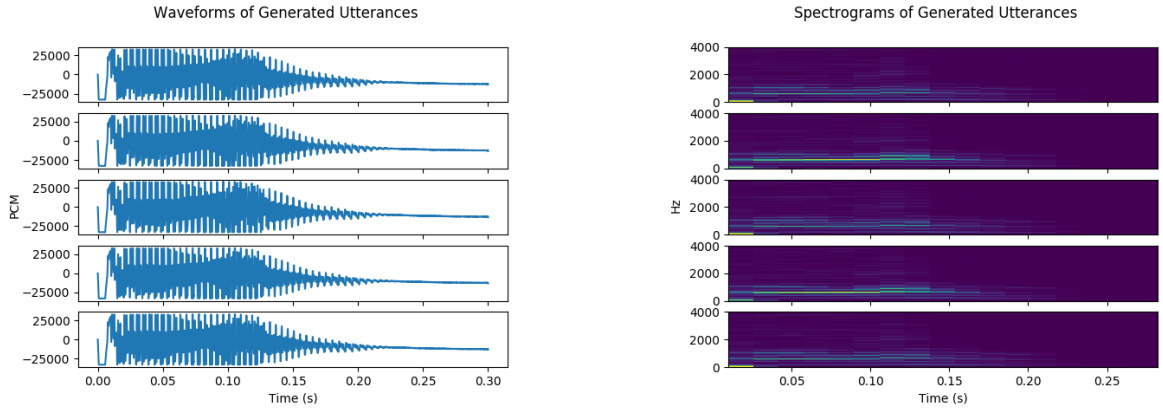

Figure 6.44: Evolution of the waveform and spectrogram of the first 0.3 second target for the genetic algorithm (12 generations, 100 population, Eucliden fitness function, 2-point crossover).

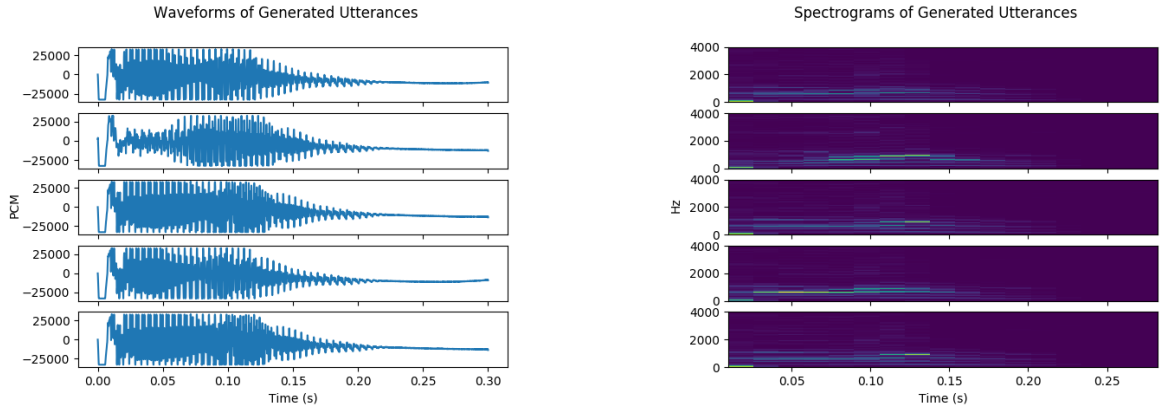

Figure 6.45: Evolution of the waveform and spectrogram of the second 0.3 second target for the genetic algorithm (12 generations, 100 population, Euclidean fitness function, 2-point crossover).

for each agent in a gene pool, the agent needs to be run through the articulatory synthesizer to generate an audio file, then the audio file needs to be read back into memory and a spectrogram needs to be generated from it, and then the fitness function needs to be applied. For the Euclidean fitness function, this involves the further step of running the spectrogram through a deep convolutional encoder to retrieve the embedding coordinates. Since this takes

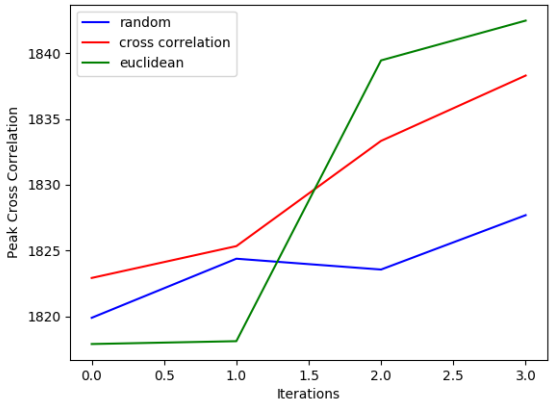

Figure 6.46: Maximum cross-correlation value over time for three different fitness functions (first target in 0.5 second condition).

so long, the number of generations and the number of agents in the gene pool were limited to 100 agents and 12 generations (for the Euclidean condition), or 25 generations (for the cross-correlation condition). This resulted in small effect sizes - the audio changes over time, just not very much.

To determine if the small changes to the sound over time are outside the realm of mere chance, the same experiments were run, but this time with a fitness function that ignored the input sound and returned instead a random number between 0 and 100. Since this should produce essentially randomly modulated pretrained synthesis outputs, we can use it to see if its variance is less than the total change we see in the previous experiments. Figure 6.46 shows just this. To construct this plot, the genetic algorithm was run using each fitness function, random, Euclidean, and cross-correlation after pretraining a pool of agents via RMS maximization. At each of five time points over the course of training, the best agent was selected (based on the value of their fitness functions). This agent was turned into a sound file and compared with the target utterance via the normalized cross-correlation procedure described previously. If, over the course of 12 generations, the Euclidean line grows outside of the value enveloped by the random line, I take this to mean that the Euclidean

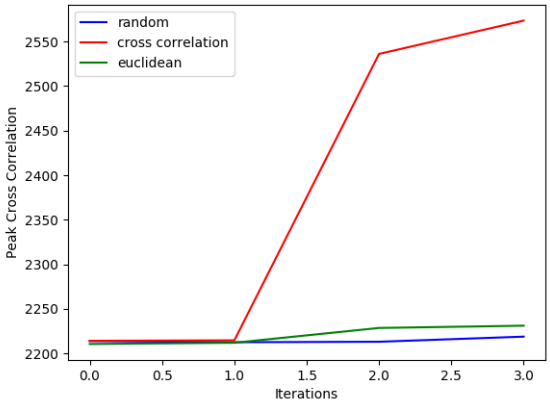

Figure 6.47: Maximum cross-correlation value over time for three different fitness functions (second target in 0.5 second condition).

fitness function is better than chance at evaluating the similarity between a sound and a target sound, and that it provides evidence in favor of Synthesis Assumption II.

It can be seen from Figure 6.46 that this is indeed the case: the Euclidean fitness function seems to produce sounds that are, over time, more similar to their targets than chance would allow. However, a word of caution is necessary: first, this is simply one sound target; second, doing this analysis with a much larger gene pool and many more generations would provide more statistical power.

To combat the first problem, I have run this analysis against all four targets: both targets from the 0.5 second condition, and both targets from the 0.3 second condition. Figure 6.47 shows the second target for the 0.5 second condition. This plot seems to show that the Euclidean fitness function barely escapes chance, and does not seem to track very well with the golden cross-correlation fitness function. Indeed, this is the case for Figures 6.48 and 6.49 as well.

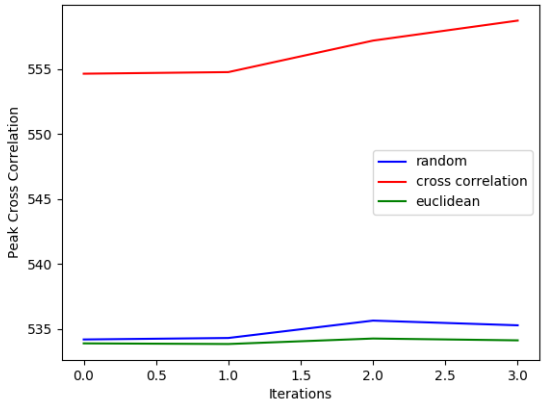

Figure 6.48: Maximum cross-correlation value over time for three different fitness functions (first target in 0.3 second condition).

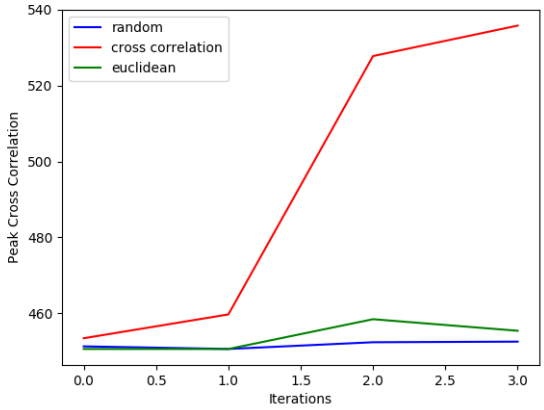

Figure 6.49: Maximum cross-correlation value over time for three different fitness functions (second target in 0.3 second condition).

6.5 Prediction: Cooing

This prediction states that the sounds that are produced as part of the first phase of the ESC model (learning to coo) will be similar to those made by infants while they are cooing. Compare Figure 6.33 with Figures 6.50,6.51, and 6.52. It is not obvious whether this has

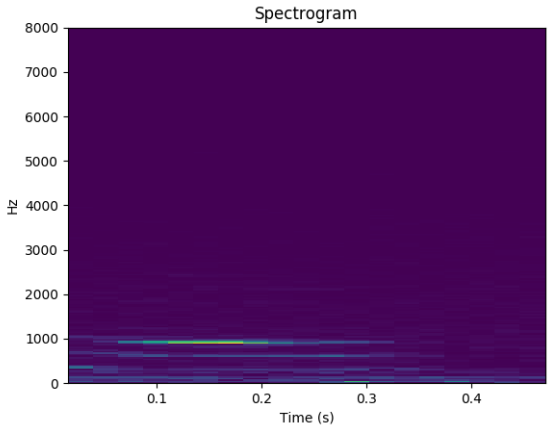

Figure 6.50: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

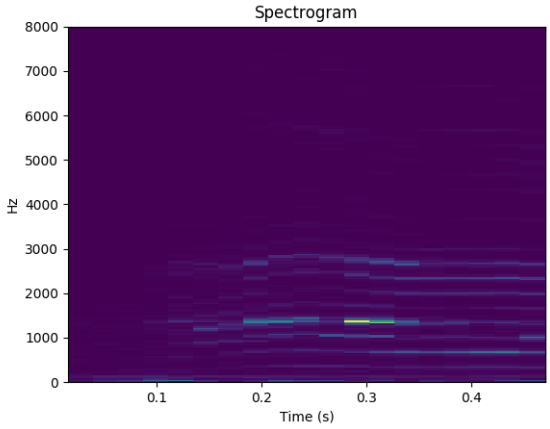

Figure 6.51: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

indeed happened. However, from listening to the sounds, it seems that the natural cooing is more varied and contains more changes to the frequency across the utterance. The simulated cooing is clearly a vowel, but does not attain the frequency shift common (and observable) in the coos. Nonetheless, the sound, were it from a child's vocal tract rather than an adult male's, would likely be passable as a coo.

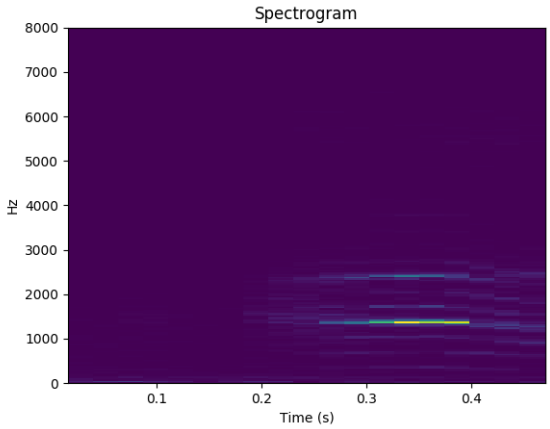

Figure 6.52: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

6.6 Prediction: Marginal Babbling and Reduplicated Babbling

This prediction states that sounds produced during the marginal babbling phase of the ESC theory will first resemble marginal and then will resemble reduplicated babbling by human infants.

Only the first half of this prediction was tested as part of this thesis: that marginal babbling will arise during this phase of the ESC model.

Figures 6.53 and 6.54 show typical examples of marginal babbling from the Oliver dataset. Compare these with the final spectrogram in 6.43. I discuss this result in more detail in the coming chapters.

6.7 Prediction: Non-Reduplicated Babbling

This prediction was not tested as part of this work.

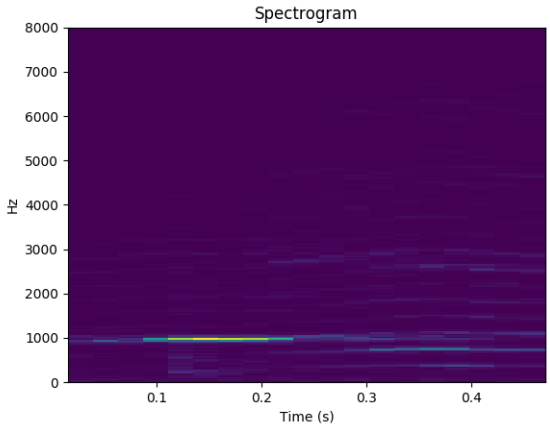

Figure 6.53: Typical example of marginal babbling from the Oliver dataset, when he was 8 months old.

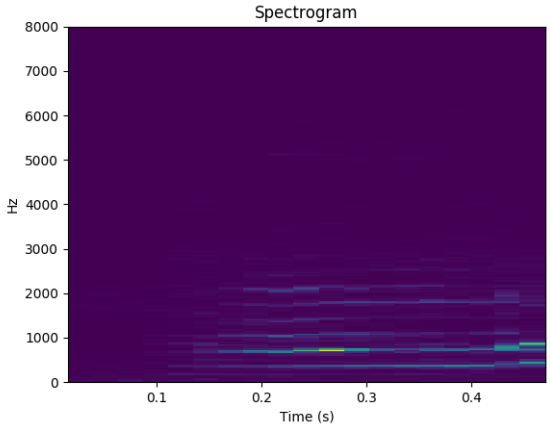

Figure 6.54: Typical example of marginal babbling from the Oliver dataset, when he was 8 months old.

Chapter 7
DISCUSSION

In this chapter, I discuss the results in more detail, including their implications. The overarching question that this chapter seeks to answer is whether the results as given in the previous chapter support the Evolving Signal Chain theory of speech acquisition. I think on the whole, the results do indeed support this new theory, and should provide a strong motivation for further development of the theory and further prediction testing, including the completion of the reference model.

7.1 Limitations

The first thing to get out of the way are the short cuts and optimizations that have been made in going from the ESC theory to the reference system specification, and then again from going to the reference system specification to an actual implementation. What were these changes and do they invalidate any results?

7.1.1 Auditory Periphery

The first difference between the ESC theory and the model I set out to create for this thesis is that the theory accounts for humans' auditory peripheries, while the model tries to abstract these away. This was done because the auditory periphery is largely important for the processing of acoustic streams in ways that are of evolutionary importance (such as startle response and sound source localization), but which are not necessarily important in language processing and acquisition. It may be the case that very low level auditory processing is important, but it is not clear why this would be the case, and as such ESC does not account for anything at this level of processing other than simple phase correction and transformation

into the frequency domain. Since these tasks can be replaced by a monoaural recording device and spectrogramification, the model should not incur any penalty in biological plausibility for excluding or discounting them.

7.1.2 Auditory Scene Analysis

The auditory scene analysis block and channel selection in ESC is important. Together, they account for the finding that infants listen to particular voices more than others and seem to prefer infant directed speech. Currently, these findings are unaccounted for in the reference system. Not only this, but because the reference system has no way of attending to one speech stream at a time, the embedding device (in the reference system an autoencoder) is faced with a more difficult task - it must learn an embedding function for not only useful speech, but also overlapping speech and non-speech noise as well. Due to this, an attempt was made to create a bottom-up auditory scene analysis block, but its results were not good enough for inclusion in the reference system. Future work may focus on this portion of the system.

Given that the ASA block and the channel detection block are so important, can we still use the reference system to test the biological plausibility of ESC? Yes. The removal of the ASA and attention blocks make for a more difficult task for the embedding function, but as long as the embedding function can still operate, we may draw the conclusion that the embedding function can work in the less difficult task that it would have if these blocks were included - and thus is a viable embedding function for ESC. In the system implemented for this thesis, the ASA and channel selection blocks were replaced by a silence filter.

7.1.3 Language Classification

The last difference is that the reference system and ESC both call for a language detection mechanism that results in different embeddings based on different languages. Such a block would likely, in the human infant, be very primitive - perhaps a rhythm detector. Although the Oliver dataset has a large amount of Mandarin and English, only English utterances were

used in the tests, while the autoencoder was trained over both. Again, this made for a more difficult task than the biological signal block is likely to face, and so if the experiments were successful, we can draw the conclusion that the embedding function would operate better in the less stringent conditions, and so declare the biological viability of the mechanism.

7.2 ESC: Speech Perception

This section discusses the results that I obtained for the experiments done to test the perception portion of the theory. I first discuss in detail the results of trying to form an embedding that handles continuous natural speech. I then discuss the implications of these results for the feasibility and biological plausibility of the ESC theory of speech acquisition.

7.2.1 Embedding Analysis

Assumption: Embedding Spaces I

This assumption states that, even if we can form an embedding space from continuous speech, phonemes will not form into obvious clusters in this embedding space. I outlined in the chapter on ESC why I believe this to be the case - mostly having to do with suprasegmental and especially coarticulatory effects. In ESC, phonemes (or whatever high-level speech commands) arise due to the hierarchical nature of the motor system, and then once formed, assist in the decoding of speech. If phonemes can be shown to arise naturally in embedding spaces of continuous speech, ESC, as currently postulated, is likely invalid, and we can opt for a simpler theory, such as NLM.

It is difficult to prove that clusters will not naturally arise in embedding spaces, and it is difficult to prove that those clusters that do arise will not correspond to phonemes as delineated by linguistic theory. However, for the reasons outlined in the chapter on ESC, it should be clear that there is no compelling reason to believe, based purely on the data to be clustered, that clusters of phonemes will form in an isolated and easily clusterable manner - a central idea behind many distributional learning theories in language acquisition.

Indeed, the evidence presented in this thesis seems to bear this conclusion out. The embeddings that were presented in the previous chapter showed few obvious clusters with a notable exception being Figure 6.15, which showed two distinct clusters in the embedding space of the 81x18x1 3D autoencoder - but this is only two clusters: a far cry from the 30+ phonemes found in the English language. Even the /a/ sounds in the previous chapter's analyses did not readily form into clusters except in one embedding space, despite spectrograms that were very similar (and reconstructed spectrograms that were more similar still). The English language vowels, which were shown in many of the plots in the previous chapter's analysis of embedding spaces, never formed into a cluster, even in one- and two-dimensional space. Indeed, even in 64-dimensional space, clusters could not be induced to form.

Given these findings and given the presence of coarticulatory effects and suprasegmental features, it seems unlikely that NLM's self-organizing maps will form clusters that categorize the unsupervised natural utterances of humans into linguistic phonemes. I therefore think we are justified in adopting a more complicated theory of language acquisition, such as ESC, and we are justified in adopting a theory that does not propose embedding spaces which learn clusters, but rather embedding spaces which are used for another purpose entirely - as a means by which an infant can form a loss function for mimicking adult speech. I come back to this later in this chapter.

Assumption: Embedding Spaces II

This assumption states that continuous natural speech can be embedded into a low-dimensional embedding at different time scales. This assumption is a key position of the ESC theory, and if I failed to find evidence in support of it, I would have to rethink the entire theory. Indeed, failing to embed continuous natural speech into an embedding space would throw all distributional learning models of language acquisition into question, since until this thesis, they have almost without exception been tested with hand-picked data, as opposed to real, continuous speech.

Based on the results reported in this thesis, I feel it is safe to say that natural speech,

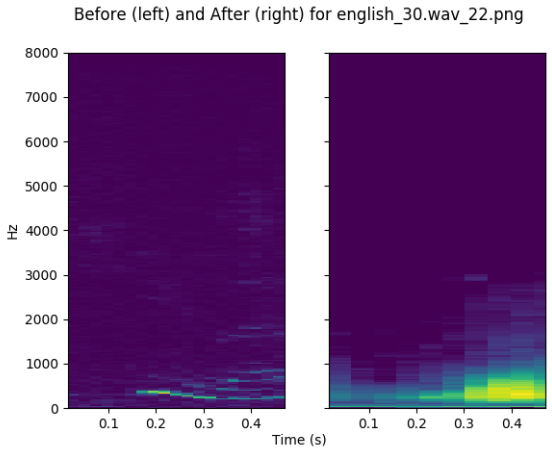

Figure 7.1: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the autoencoder's output is on the right. Part one of two.

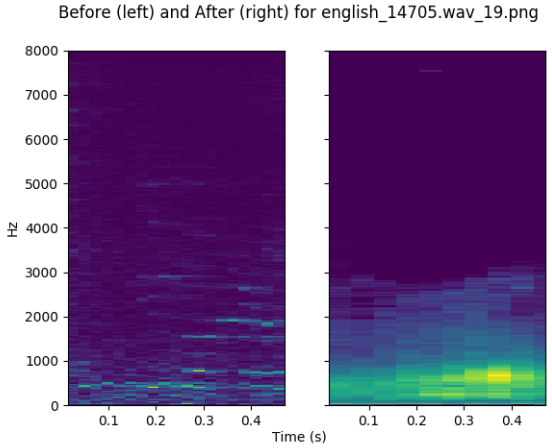

Figure 7.2: Typical spectrogram reconstruction for the 241x20x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the autoencoder's output is on the right. Part two of two.

buffered at different time scales, can be encoded into a low dimensional embedding space.

Figures 7.1, 7.2, 7.3, and 7.4 show the reconstruction of input spectrograms for two different targets in each of the two different spectrogram resolutions (these figures are the same as Figures 6.1, 6.2, 6.3, and 6.4 from the previous chapter).

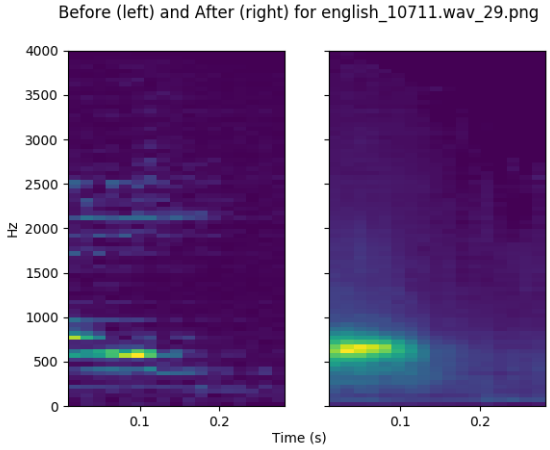

Figure 7.3: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the autoencoder's output is on the right. Part one of two.

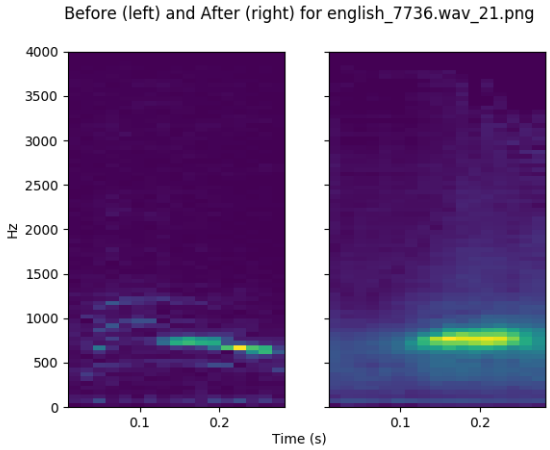

Figure 7.4: Typical spectrogram reconstruction for the 81x18x1 vanilla convolutional autoencoder. Input spectrogram is on the left, while the autoencoder's output is on the right. Part two of two.

It can be seen from inspection that, at least for the spectrogram and embedding dimensionalities reported here, that the embedding space must contain at least some (and I would argue *most*) of the information in the time slice of the speech utterance. I believe that, with a better autoencoder architecture and training regimen, these results could be made

even more accurate at reconstruction. But the accuracy of the reconstruction is only really important as far as learning a reasonable embedding space is concerned. For this purpose, the results presented here should suffice.

7.2.2 Biological Plausibility of Embedding Speech Utterances

The assumptions for the perception portion of the ESC theory have been born out by the work done in this thesis, but what about the biological plausibility of this portion of the theory? The chapter on ESC has already lain out the findings in linguistics and psychology that have motivated this theory, so I will not revisit these motivations here. Instead, I will discuss how the information gleaned from the analyses in the results chapter informs whether or not this theory is viable.

First, ESC relies on having several embeddings of natural speech - one for many different time scales (tested in this thesis at 0.3 seconds and 0.5 seconds), and groups of these per language category. Based on the results of this thesis, does this seem reasonable? The answer seems to me to be yes. A deep convolutional autoencoder is capable of forming meaningful embedding spaces for at least 0.3 seconds and 0.5 seconds, with no reason to suspect that these results could not be extended to shorter or longer (within reason) time scales. Having said that, I should note that the 0.3 second autoencoder architecture did seem to do better at reconstruction (and therefore at learning a good embedding of the data). This is unsurprising, since longer spectrograms will encode more information, which will result in a more difficult task for the autoencoder. Obviously, at some time scale, it becomes impossible to learn an embedding. But humans clearly do not have an arbitrarily long buffer for speech. So one natural question is how long a human's audio buffer is, and whether the amount of information present in continuous speech for that length of time can be successfully encoded into a low-dimensional embedding space. If not, ESC would need to be adjusted somehow to account for this finding. Of course, humans probably do not buffer audio in the sense that ESC does - the nervous system instead will feature many neurons which take in activity at different frequencies and either fire or decay back to the resting

potential based on upstream activity in the signal chain. Thus, figuring out the appropriate analog for ESC's audio buffer would be the first step in testing this.

A second thing to talk about is the role that embedding spaces play in ESC. In typical distributional learning theories of language acquisition, the role of the embedding spaces is to form a space that has a number of dimensions that is conducive to clustering algorithms (whatever those may be in the human brain - typically these theories are tested using self-organizing maps). But my findings here, and my arguments in the chapter on ESC show that such clustering may not occur at all - and certainly it is not easy to induce them, as long as continuous natural speech is used, as opposed to hand-parsed or syntheiszed vowels. If this is the case, then what use do we have for embedding spaces? The answer to this question is that rather than perform a clustering in the low-dimensional embedding spaces, we use these spaces for optimization. That is, once we have an internal representation of speech, we can use this representation as a means to evaluate the similarity between our own utterances and the utterances that we have heard. In this thesis, I tested the idea that minimizing the distance between the embedding of a target utterance and the embedding of an infant's attempt at mimicking that utterance will result in more and more similar mimicry over time.

What about categorical perception? NLM and other distributional learning theories of speech acquisition account for this phenomenon by means of the clustering that I have just discarded. How then can ESC account for the fact that adult speakers perceive vowels as belonging to particular categories, even when they are slightly (or sometimes very) different, quantitatively? I argue, as other DL theories do, that categorical perception arises due to a phoneme or phoneme-like embedding, but rather than this embedding existing as a result of clustering perceived speech, it is instead a result of the speech production system. This idea borrows a page from the motor theory of speech recognition, albeit a much weaker version of it than has been expounded by its proponents throughout the years. The idea is this: infants, as they learn to map from sounds to articulator control, learn an embedding that does result in easy clustering. If we were to collect the articulatory control matrices for the Praat synthesizer for several variations of the sound [a], and for several variations of the

sound [e], and again for [m], etc. - as used in words where coarticulatory effects are present - and if we applied an unsupervised clustering algorithm to the space of all these matrices, would we encounter clusters of phonemes? I do not know, but I think that if we were to embed this large-dimensional space into a smaller one in such a way as to maximize acoustic reconstruction (that is, in such a way so as to not lose our ability to articulate properly when we decode the embeddings), we would find phoneme clusters. In this way, the brain would form phonemes by minimizing the amount of control over articulation, while still maintaining the ability to produce all the sounds that are required of it. Therefore another prediction of the ESC theory of speech acquisition is that categories can be induced in an embedding space that is hooked up to the speech synthesizer, as opposed to the speech perception module.

I should note that while I believe that *phonemes*, if they form in the brain at all, do so in the way just outlined, this does not change the fact that humans are capable of being trained to perceive categories, as is the case when infants are trained to perceive a unimodal or bimodal distribution of noises, as in [32]. There is clearly evolutionary advantage to heuristic ways of thinking, and categorical perception is merely one mechanism by which heuristics can be formed. I therefore do not take the view that infants cannot show categorical perception effects before they are able to control their articulators, I merely think that they will not be sensitive to phoneme boundaries (without laboratory training) until they are old enough to start controlling them.

Now let's turn our attention to a question that I have ignored so far, but which is important for deteriming biological plausibility - what exactly is an embedding space in the human brain? I have shown that speech sounds can be represented, with some loss of information, by vectors as small as two dimensions, and prosody can probably be represented in short time scales in one-dimension. Increasing the number of dimensions in the embedding space results in more information being retained.

Figures 7.5 and 7.6 show the principal component analysis (PCA) for the test set of the 241x20x1 spectrogram condition and the 81x18x1 spectrogram condition, respectively. It is interesting to note how little variance in the dataset can be captured by a single axis.

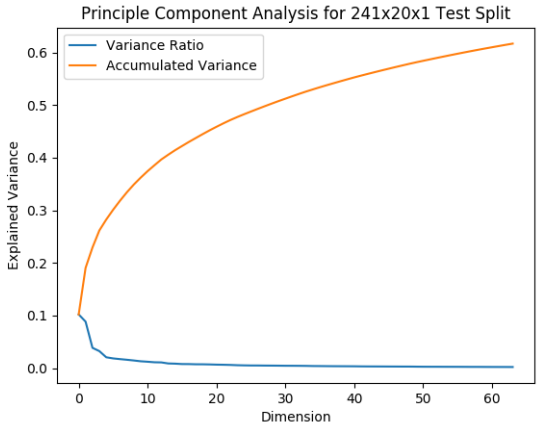

Figure 7.5: Principal Component Analysis of the test split for the 241x20x1 (0.5 second) spectrogram condition. The blue line indicates the amount of variance explained by each axis of the resulting projection, while the orange is the accumulated explained variance.

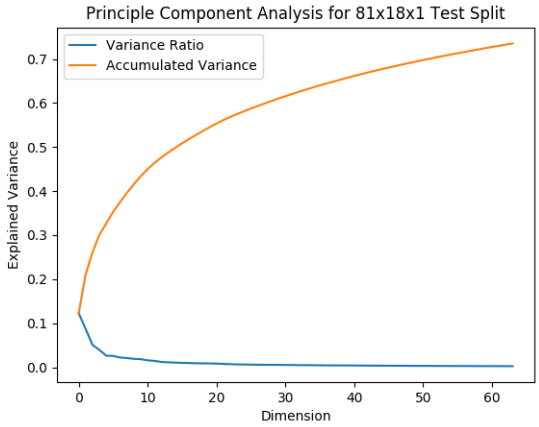

Figure 7.6: Principal Component Analysis of the test split for the 81x18x1 (0.3 second) spectrogram condition. The blue line indicates the amount of variance explained by each axis of the resulting projection, while the orange is the accumulated explained variance.

Indeed, over 64 dimensions, not even 80% of the variance is captured. Viewed in this light, it is surprising how well the autoencoders in three dimensions did at learning a faithful embedding space. A few possible interpretations of embedding spaces exist. The first is that

in the embedding space, each dimension corresponds to a physical dimension. This would be somewhat akin to a self-organizing map. For this to be possible, we would need to constrain our embedding spaces to dimensionality less than or equal to three. I see no good reason for this to be the correct interpretation. The second interpretation would be that there exists some neural architecture in the brain that encodes the utterances into N literal axons. This would be the interpretation that is most faithful to an autoencoder artificial neural network. If this is the case, we may choose numbers that range into the thousands for our embedding spaces' dimensionalities, as the human brain has tens of millions of axons. There are yet other reasonable interpretations, but the point here is simply that there is no reason not to use dimensionalities of almost arbitrary size - something that may be necessary, given the PCA figure shown above.

One last thing to mention here is that ESC, as formulated in this thesis, does not currently specify with enough detail how the embedding spaces affect one another or how they change over time. If pressed for an opinion today on how they should coordinate, I would say that they should take on a TRACE-like mechanism, where shorter time scales inform longer ones as another input into that time scale's autoencoder, and longer time scales can inhibit shorter time scales. Such a neural architecture has elegance and is intuitive, but would unfortunately be quite onerous to get right and test as a computer model. Time constraints prevent me from both defining this portion of the theory and testing it. I therefore leave this portion of the theory underspecified and this portion of the reference system completely unspecified - it shall therefore be implementation defined for the time being.

7.3 ESC: Speech Production

7.3.1 Production Analysis

This section discusses the results I obtained for the speech production portion of the theory.

Assumption: Synthesis I

This assumption states that there exists some algorithm by which we can produce noticeably different speech sounds. The reason for this assumption is that without some way of making speech, the ESC theory cannot be modeled. And since computational modeling is one of the central tenets of ESC, this is important.

The Praat articulatory speech synthesis model was used in this thesis as the software component for speech synthesis. It was chosen because it was one of the few available *articulatory* speech synthesizers, and was the only one that I could find that could easily be hooked into a Python program.

The results of testing this were surprising. Although Praat's articulatory synthesizer can indeed be made to produce noticeably different utterances, it is non-trivial to do so. Nonetheless, it was important to me that I use an articulatory speech synthesizer, since it was unclear at the outset of this thesis how much importance should be assigned to the realism of the production mechanism, and articulatory synthesizers are the most realistic voice synthesizers available. While this choice was the right one, given the information I had at the time, I have come to believe that the importance of the particulars of the synthesizer are not currently important to the theory of ESC. This may change in the future, as I hope to expand ESC to include more linguistic development, and indeed, more development in general. But for now, the disadvantages of using an articulatory synthesizer - mostly the complexity of existing models and lack of diversity in open source implementations - outweigh the advantage of biological realism. Future work may involve determining a better speech synthesizer component.

Assumption: Synthesis II

This assumption states that speech utterances can be made more similar to one another by minimizing the distance between them in an embedding space. This assumption is critical to ESC, as it is the hypothesized mechanism by which infants learn to make sounds that

are similar to those they hear in their linguistic environment. If this assumption is proven invalid, it would shake the foundations of the ESC theory.

The experiments that were run to test this assumption were hampered by a few problems. First, the Praat articulatory synthesizer presents a very high dimensional search space - there are 27 distinct articulators (not including the lungs), each of which must be set to a floating point value for several time points in order to produce an utterance. This results in control matrices that are, in most of my experiments, $27 * 3 = 81$ dimensions in size. Not only this, but the interactions between the different articulators are non-linear. I have tried to deal with this by zeroing the values for many of the less important articulators, and by annealing the limits of different articulator groups at different times in training, but the search space remains quite large. It is important to note however, that this is exactly the same problem in articulation that infants face as well - infants have an enormous number of muscles, almost none of which can be controlled voluntarily at birth, and they interact in non-linear ways.

The second problem I faced in testing this assumption is that there is no great method to use to train a simulator to use the Praat articulators. Ideally, since the problem is posed as a supervised learning process (we have embeddings from the speech database and we have a loss function: the euclidean distance between guessed embeddings, $\hat{\mathbf{Y}}$ and the targets, $\mathbf{Y}$), we could use a powerful supervised learning process, such as a deep neural network. But unfortunately, this is not the case. In order to use supervised learning, I need a database of the form ($\mathbf{X}$, $\mathbf{Y}$), where each $\mathbf{y}$ in $\mathbf{Y}$ is paired to each $\mathbf{x}$ in $\mathbf{X}$ such that I want to learn a function that takes any $\mathbf{x}$ and returns the corresponding $\mathbf{y}$. But since the $\mathbf{y}$ that I want is a control matrix, this would require that I have the control matrices already. I do not. And there is no easy way to make them. Thus, I have had to rely on a genetic algorithm. But my genetic algorithm is dealing with an almost certainly non-convex fitness function, and I can't introduce enough genetic variability and long enough simulations to deal with this appropriately because of computational constraints.

However, despite these limitations, the results obtained from these experiments are promising. In the experiments used to test this assumption, I have made a further assump-

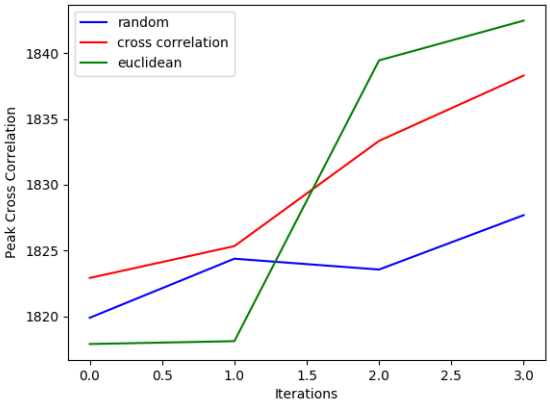

Figure 7.7: Maximum cross-correlation value over time for three different fitness functions (first target in 0.5 second condition).

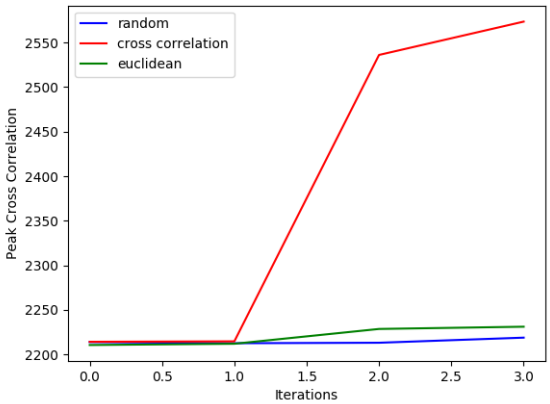

Figure 7.8: Maximum cross-correlation value over time for three different fitness functions (second target in 0.5 second condition).

tion (though one that is mathematically motivated): that the maximum value of the cross correlation between two waveforms is a good stand-in for the vaguer notion of the similarity of two speech utterances. After making this assumption, I tested the intermediate results of the Euclidean fitness function against this standard: i.e., I took the maximum value of the cross-correlation between the intermediate waveform and the target for each experiment

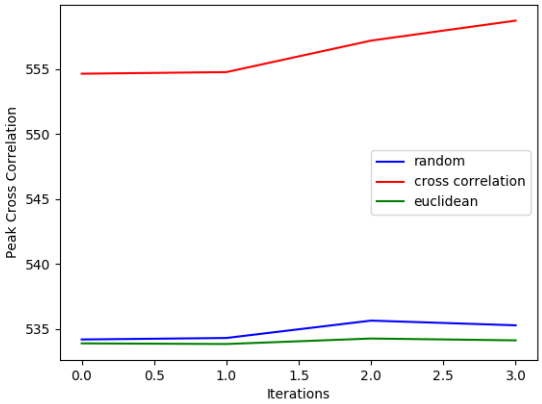

Figure 7.9: Maximum cross-correlation value over time for three different fitness functions (first target in 0.3 second condition).

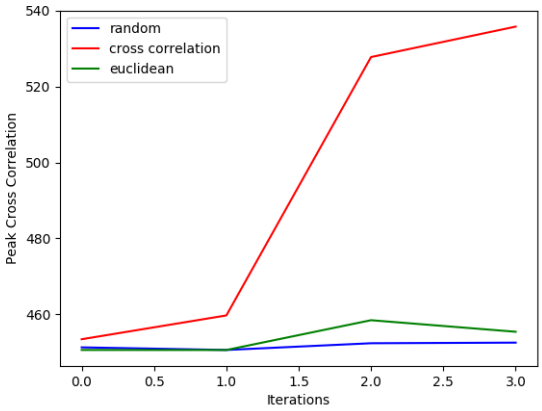

Figure 7.10: Maximum cross-correlation value over time for three different fitness functions (second target in 0.3 second condition).

at several points during the experiment. I then plotted these values, along with the values obtained by using this cross-correlation procedure as a fitness function in its own right, and the cross-correlation values of randomly generated utterances. These figures were shown in the previous chapter as Figures 6.46, 6.47, 6.48, and 6.49, and I reproduce them here as Figures 7.7, 7.8, 7.9, and 7.10.

In the previous chapter, I noted that in all but one of these plots, we can say that the Euclidean distance function resulted in gains that were outside the realm of chance, though in two of those three, this was only barely. I should note Figure 7.9, where the Euclidean distance fitness function did not escape chance improvement, shows only a slight gain by the cross-correlation ("gold standard") fitness function. This indicates to me that the similarity of a Praat-generated utterance and the first target in the 0.3 second condition was simply difficult to maximize.

Overall, the reults show statistical significance, though with low statistical power. To be able to safely say that this assumption is validated would require that I run these experiments with much higher genetic variability and for much longer. Ideally, I would then do that with several different random seeds. Alternatively, I could explore the use of some other optimization framework. This is an avenue of future research.

7.3.2 Biological Plausibility of Using Embeddings for Speech Mimicry

As discussed earlier in this chapter, ESC departs from typical distributional learning theories in the way that it makes use of its embedding space. Rather than hypothesizing the learning of an embedding space as a means by which we can cluster around phonemes, ESC hypothesizes the existence of embedding spaces as a means by which we may optimize mimicry in a lower-dimensional space. A central assumption of ESC therefore is that such an embedding space can be used for this purpose. As mentioned in the previous subsection, results are promising for validating this assumption, though not quite up to the task of doing so definitively. Further research into the suitability of embedding space topographies for optimization is needed before I can say with certainty that this method of learning to mimic speech is viable.

At the beginning of this chapter, I mentioned the limitations of the computer model used to test the ESC theory. One of these limitations was that Mandarin, English, and non-speech noise were all lumped together into the dataset that the autoencoders had to deal with. This placed additional burden on the autoencoders, on top of an already difficult task. The fact

that they were able to do well is surprising and pleasing - as it lends credence to the feasibility of learning embeddings of continuous natural speech. But this raises an interesting question: how much better would the autoencoders do if the voice-pass filter and language classifiers were implemented successfully? And how would this affect the topography of the embedding spaces insofar as is important for the mimicry loss function?

In this same vein is another question: what would happen if we implemented tha ASA block? If we trained the autoencoders on speech only if that speech had good signal to noise ratio, or if the training regimen strongly favored infant-directed speech, how might the embedding spaces be affected? Perhaps the embedding space would be more suitable for optimization.

Untested in this thesis is the hypothesis that infants use cargiver utterances as corrective signals. A question central to that hypothesis is what mechanism this supervision might adopt. It seems to me that the answer to this would be a sequence of events like this: infant utters [əlʃ], caregiver says "That's right, *us*" ([ðætsɹait <pause> ʌ:s:]), infant's volitional block takes in sensory and temporal context and "realizes" that a caregiver is correcting it, and then it adjusts the learning mechanism such that the infant does not adjust its parameters in order to minimize the distance between its utterance and whatever embedding it was targetting, but rather, it attempts to adjust its parameters so that it minimizes the distance between the caregiver's embedding and the recent utterance's embedding. Testing this would require a mechanism to realize when speech is meant to be corrective, and then a way to parse out the corrective portion of that speech. I currently do not specify these mechanisms. In this scenario, the caregiver pauses significantly before the corrective utterance, and enunciates it using infant-directed speech, and from personal experience, this seems like a robust pattern for corrective utterances, but I have not researched this.

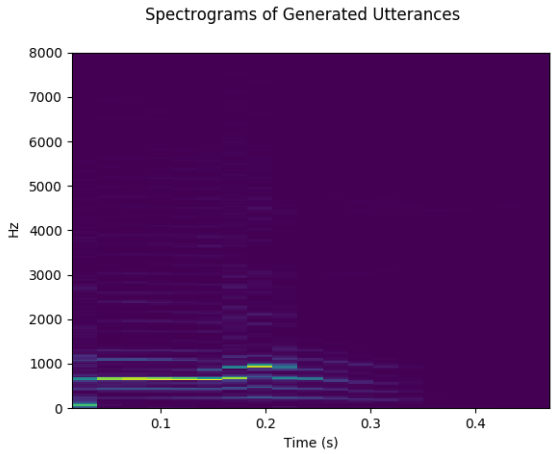

Figure 7.11: Typical spectrogram of the best agent from a gene pool trained to maximize RMS (0.5 second utterance).

7.4 ESC: Combined System

Prediction: Cooing

This prediction states that the sounds that are produced as part of the cooing phase of the ESC model will be similar to those made by infants while they are cooing. To test this, the sounds from the pretraining of the genetic algorithm (those produced from maximizing RMS) were plotted as spectrograms and compared against some typical coos found in the Oliver dataset.

Figure 7.11 is a typical sound produced by the RMS-maximization algorithm, while Figures 7.12, 7.13, and 7.14 are typical examples of cooing taken from the Oliver dataset at around 12 weeks of age.

Qualitatively, the sounds are similar. They are typically dominated by a single frequency and its harmonics. The main difference is that the artificial sound has fewer harmonics.

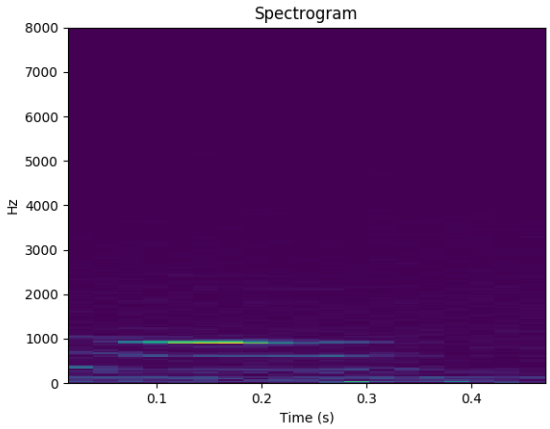

Figure 7.12: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

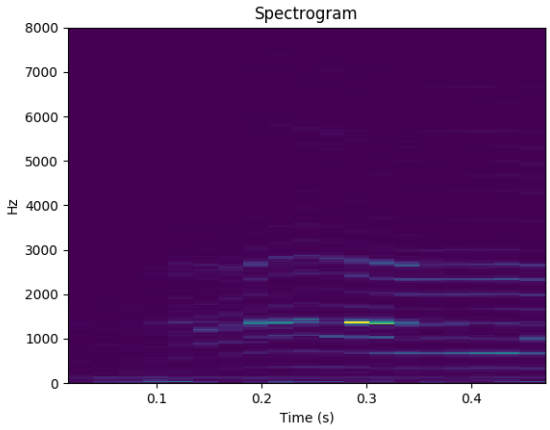

Figure 7.13: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

Prediction: Marginal Babbling and Reduplicated Babbling

Marginal babbling is created by modulating coos into varied vowel-like sounds. Figures 7.15 and 7.16 show typical marginal babbling in the Oliver dataset, at about the age of 8 months. The sounds are more varied than the sounds produced by ArtieInfant. To really achieve marginal babbling would require that I run the genetic algorithm for much longer than is feasible. Again, an avenue for future research is how best to address the problem of optimizing

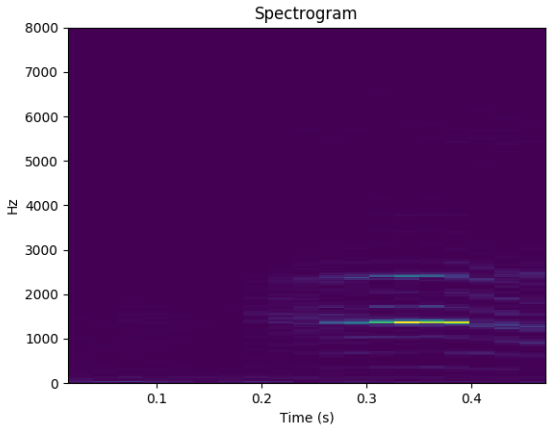

Figure 7.14: Typical example of cooing from the Oliver dataset, when he was 12 weeks old.

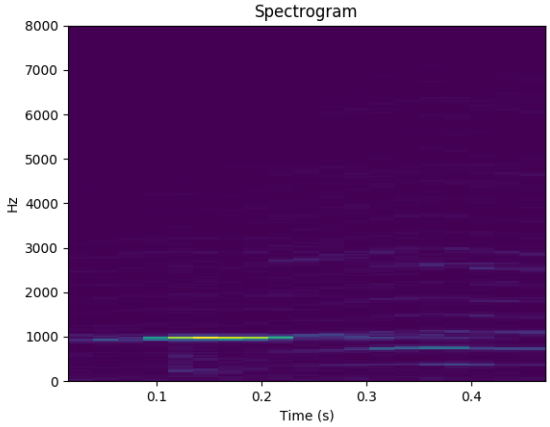

Figure 7.15: Typical example of marginal babbling from the Oliver dataset, when he was 8 months old.

the articulatory synthesis model. Future work will revisit this prediction.

Prediction: Non-Reduplicated Babbling

This prediction was not tested as part of this work.

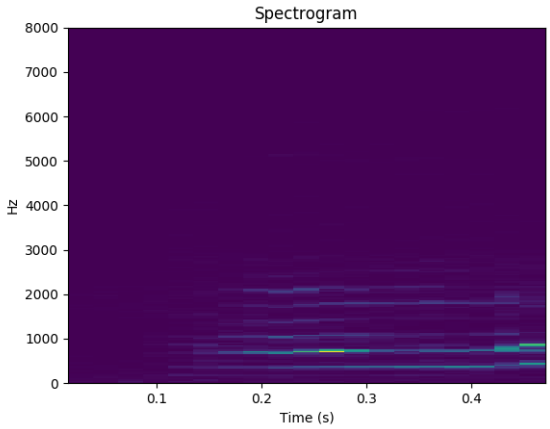

Figure 7.16: Typical example of marginal babbling from the Oliver dataset, when he was 8 months old.

7.5 ESC: Final Notes

Overall, the Evolving Signal Chain theory of language acquisition seems to have motivation due to the evidence presented in this thesis. This theory still needs some work, especially in regards to the recurrent neural network and how the embeddings grow and coordinate with one another, and of course, in exactly the mechanism used by the volitional block to bring about change in the signal chain, but in sum, it should provide new avenues of research and new ideas in the study of language acquisition.

A final note on the question of biological plausibility insofar as ESC is concerned: is it the case that, from the results analyzed here, we can expect a fully realized ESC reference system to go through the correct time course? I.e., can we expect first cooing, then marginal babbling, then reduplicated babbling, then non-reduplicated babbling, then prosody, then whole words? Clearly, the referene system learns to coo first, which is here modeled as simply learning to maximize the RMS of utterances (and thus learn a good starting point for further utterances in later stages). Unfortunately, the recurrent neural network could not be implemented due to time constraints, and therefore I do not know if the timeline would have been obeyed. It seems likely that, if we enforce the volitional block to sample

from the short time scales until we can successfully produce simple sounds, and then allow the volitional block to sample from longer time scale embeddings, we would not only do a good job bootstrapping the difficult task of training the neural network, but we would also follow the expected timeline. This is an exciting idea, since it lends strong support for this mechanism and the theory of ESC in general. Future work may therefore focus on this aspect of ESC.

Chapter 8

CONCLUSION

In this thesis, I presented a new theory of primary phonology acquisition, the Evolving Signal Chain theory, which may be further expanded to primary language acquisition in general. The motivation for this theory comes from a lack of current theories in language acquisition to adequately account for all research into this field and especially a lack of theories specific enough to withstand end-to-end computer simulation. Based on a thorough review of the current literature, ESC seems to account for the most robust findings in infant language acquisition research.

In order both to prove the testability of ESC and to actually test key parts of it, I specified a reference system, which I implemented as faithfully as could be expected, given the ambitious nature of this work and the timeline faced. Certain components still require implementation and testing, but the key assumptions of ESC were successfully tested as part of this work. The results of this testing showed that the theory of ESC seems well-suited for explaining observations in studies of language acquisition in infants.

This thesis represents original work that contributes to the fields of language acquisition research and artificial intelligence in general. Specifically, this is the first theory that I know of that is end-to-end testable with a computer model: i.e., it describes in detail how an infant goes about learning the phonology of their native language(s). Secondly, this thesis presents the first model that I know of that takes as input continuous natural speech, as spoken in an environment encountered by a real human infant. Thirdly, this thesis presents the first model of speech acquisition that I am aware of to make use of deep learning techniques, and an autoencoder instead of a self-organizing map. Lastly, this thesis presents the first time that I am aware of that a computational model has tested the feasibility of learning to mimic

speech from minimizing the distance between a target utterance and a produced utterance in a learned embedding space. I am not the first to propose this mechanism of self teaching, but I have not seen another study that has actually tested it with an embedding space that was not specified *a priori*.

8.1 Predictions

The Evolving Signal Chain theory of phonology acquisition makes several predictions that could not be tested as part of this work due to time constraints. First, what is the length of the audio buffer in humans? Clearly humans cannot retain sound forever - what is the longest biologically plausible spectrogram that must be embedded? ESC predicts that continuous speech of that length can be embedded into a low-dimensional space. If this is not the case, it will be necessary to revisit whether embeddings at that length are truly necessary, and if not, why not?

Second, ESC predicts that if phonemes exist, they will only arise as a product of the hierarchical nature of the human motor system, not as a result of learning to decode a speech stream. ESC allows for the use of these phonemes in decoding speech, but it does not require it. If it is the case that humans learn phonemes before they learn to say them, key parts of ESC would need to be reconsidered.

Last, ESC predicts that in training a recurrent neural network to sample from and successfully reproduce utterances in the shorter time scale embedding spaces and then longer time scales will produce a timeline of spontaneous utterances that closely resembles that of infants' babblings - i.e., cooing, then marginal babbling, reduplicated babbling, non-reduplicated babbling, prosody, and then finally words. If this does not pan out, there will be a question of how exactly this time course should be realized.

8.2 Future Work

This thesis is complete, but in doing the research for it, I have discovered many areas that would benefit from further exploration. Since I have time constraints, I cannot explore them

here, but I can at least enumerate them.

First, ESC was originally conceived with a requirement that it not only be end-to-end testable, but also that it be end-to-end real-time attainable. This requirement was dropped, as it was not important enough to the underlying theory and represented too hard of an engineering challenge for the research timeline. But I still believe that the end goal for any theory of developmental psychology should be, if we want artificial general intelligence, to realize the theory in an embodied, real-time system. Anything short of this fails to account for the reality in which humans live and have evolved. As such, one avenue for future research is to embody ESC into a robotic platform. This would be in good company: there is a field of psychology and artificial intelligence research called developmental robotics that has as a core tenet the idea that embodiment is central to intelligence as we understand it. Unfortunately, ESC may not yet be at a point where it could benefit from this. Embodiment would allow for hard real-time experiments, but these experiments would require an enormous amount of data - how many hours of speech is required before the autoencoder is trained at even the shortest time scale?

More important than embodiment is simply finishing the reference system. Specifically, the auditory scene analysis block and attention mechanism (or at least the voice-pass filter and language classifier) could unlock several more interesting areas of research into the biological plausibility of ESC. Elucidating and implementing the mechanism by which caregiver supervision happens or by which the different embedding spaces coordinate and change over time would also be of benefit in determining the viability of ESC. Finally, implementing the recurrent neural network and testing the prediction that the result will be similar to the timeline of spontaneous speech utterances found in human infants is also important.

Another area for research is the motor system. Little research was done as part of this thesis into how exactly neurons are recruited to particular motor tasks, nor was much research done into how this changes over time in human infants. But these are important questions, as the articulatory synthesis model in this thesis was very inefficient, and knowing more about how an infant's motor system changes over time to control the articulators might allow me

to simplify the synthesis model without loss of generality.

More specific things that could be tried are using a mel or gammatone frequency binning algorithm for spectrogram formation, rather than the linear one I used in this thesis. Such an algorithm is more biologically realistic, and may therefore be more suited to learning embeddings from data dominated by productions from the human vocal tract. Another option for increasing the realism of the reference system would be to reimplement any of it using spiking neural networks.

Lastly, I could explore intrinsic motivation as a mechanism for scheduling training regimens via the volitional blocks in the signal chain. A good example of such research is [39].

BIBLIOGRAPHY

[1] Jessica S Arsenault and Bradley R Buchsbaum. Distributed neural representations of phonological features during speech perception. 35(2).

[2] International Phonetic Association. Ipa chart, 2015. [Online; accessed 11-February-2019; Available under a Creative Commons Attribution-Sharealike 3.0 Unported License.].

[3] Paul Boersma and David Weenink. Praat: doing phonetics by computer, 2017.

[4] Kathy M. Carbonell and Andrew J Lotto. Speech is not special...again. *Frontiers in Psychology*, 5, 2014.

[5] Robin Panneton Cooper and Richard N. Aslin. Preference for infant-directed speech in the first month after birth. *Child Development*, 61(5):1584–95, 1990.

[6] C.T. Ellis and N.B. Turk-Browne. Infant fmri: A model system for cognitive neuroscience. *Trends in Cognitive Sciences*, 22(5):375–387, 2018.

[7] Tiffany Field, Lisa Guy, and Vivian Umbel. Infants' responses to mothers' imitative behaviors. *Infant Mental Health Journal*, 6(1):40–44, 1985.

[8] Bruno Galantucci, Carol Fowler, and M. Turvey. The motor theory of speech perception reviewed. *Psychonomic Bulletin & Review*, 13(3):361–377, 2006.

[9] Roberta Michnick Golinkoff, Dilara Deniz Can, Melanie Soderstrom, and Kathy Hirsh-Pasek. (baby)talk to me: The social context of infant-directed speech and its effects on early language acquisition. *Current Directions in Psychological Science*, 24(5):339–344, 2015.

[10] Ian Goodfellow. Deep learning, 2016.

[11] Jordan R. Green, Christopher A. Moore, and Kevin J. Reilly. The sequential development of jaw and lip control for speech. *Journal of Speech, Language, and Hearing Research*, 45(1):66–79, 2002.

[12] Frank H. Guenther and Tony Vladusich. A neural theory of speech acquisition and production. *Journal of Neurolinguistics*, 25(5):408–422, 2012.

[13] Gesa Hartwigsen, Annette Baumgaertner, Cathy J. Price, Maria Koehnke, Stephan Ulmer, and Hartwig R. Siebner. Phonological decisions require both the left and right supramarginal gyri.(PSYCHOLOGICAL AND COGNITIVE SCIENCES)(author abstract)(clinical report). 107(38).

[14] Erika Hoff. *Language Development*. Cengage Learning, jan 2013.

[15] Derek M. Houston and Peter W. Jusczyk. The role of talker-specific information in word segmentation by infants. *Journal of Experimental Psychology: Human Perception and Performance*, 26(5):1570–1582, 2000.

[16] Ian S. Howard and Piers Messum. Learning to pronounce first words in three languages: An investigation of caregiver and infant behavior using a computational model of an infant.(research article). *PLoS ONE*, 9(10), 2014.

[17] G. Hu and D. Wang. Auditory segmentation based on onset and offset analysis. *IEEE Transactions on Audio, Speech and Language Processing*, 15(2):396–405, 2007.

[18] John C. L Ingram. Neurolinguistics : an introduction to spoken language processing and its disorders, 2007.

[19] Peter W. Jusczyk and Richard N. Aslin. Infants' detection of the sound patterns of words in fluent speech. *Cognitive Psychology*, 29(1):1–23, 1995.

[20] Katrin Kirchhoff and Steven Schimmel. Statistical properties of infant-directed versus adult-directed speech: Insights from speech recognition. *Journal of the Acoustical Society of America*, 117(4):2238–46, 2005.

[21] Eric I. Knudsen. Supervised learning in the brain. *Journal of Neuroscience*, 14(7):3985–3997, 1994.

[22] T. Kokkinaki and G. Kugiumutzakis. Basic aspects of vocal imitation in infant-parent interaction during the first 6 months. *Journal of Reproductive and Infant Psychology*, 18(3):173–187, 2000.

[23] P K Kuhl and A N Meltzoff. Infant vocalizations in response to speech: vocal imitation and developmental change. *The Journal of the Acoustical Society of America*, 100(4 Pt 1), 1996.

[24] Patricia K. Kuhl. Learning and representation in speech and language. *Current Opinion in Neurobiology*, 4(6):812–822, 1994.

[25] Patricia K. Kuhl, Jean E. Andruski, Inna A. Chistovich, Ludmilla A. Chistovich, Elena V. Kozhevnikova, Viktoria L. Ryskina, Elvira I. Stolyarova, Ulla Sundberg, and Francisco Lacerda. Cross-language analysis of phonetic units in language addressed to infants. *Science*, 277(5326):684–686, 1997.

[26] P.K. Kuhl, B.T. Conboy, S. Coffey-Corina, D. Padden, M. Rivera-Gaxiola, and T. Nelson. Phonetic learning as a pathway to language: New data and native language magnet theory expanded (nlm-e). *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):979–1000, 2008.

[27] Alexander LeNail. Nn-svg. [Online; accessed 25-May-2019].

[28] v. Leon. Stack exchange: How do i draw an lstm cell in tikz? `https://tex.stackexchange.com/questions/432312/how-do-i-draw-an-lstm-cell-in-tikz`, 2018. [Online; accessed 02-March-2019].

[29] A. M. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdert-Kennedy. Perception of the speech code. *Psychological Review*, 74(6):431–461, 1967.

[30] Alvin M. Liberman and Ignatius G. Mattingly. The motor theory of speech perception revised. *Cognition*, 21(1):1–36, 1985.

[31] Frederic Martini. Essentials of anatomy & physiology, 2010.

[32] Jessica Maye, Janet F. Werker, and LouAnn Gerken. Infant sensitivity to distributional information can affect phonetic discrimination. *Cognition*, 82(3):B101–B111, 2002.

[33] James L Mcclelland and Jeffrey L Elman. The trace model of speech perception. *Cognitive Psychology*, 18(1):1–86, 1986.

[34] Jacques Mehler, Josiane Bertoncini, Michele Barriere, and Dora Jassik-Gerschenfeld. Infant recognition of mother's voice. *Perception*, 7(5):491–497, 1978.

[35] Piers Messum and Ian S. Howard. Creating the cognitive form of phonological units: The speech sound correspondence problem in infancy could be solved by mirrored vocal interactions rather than by imitation. *Journal of Phonetics*, 53(C):125–140, 2015.

[36] Daniel Mirman, Lori L. Holt, and James L. McClelland. Categorization and discrimination of nonspeech sounds: Differences between steady-state and rapidly-changing acoustic cues. *Journal of the Acoustical Society of America*, 116(2):1198–1207, 2004.

[37] Holger Mitterer, Odette Scharenborg, and James M. Mcqueen. Phonological abstraction without phonemes in speech perception. 129(2):356–361.

[38] Monika Molnar, Judit Gervain, and Manuel Carreiras. Within-rhythm class native language discrimination abilities of basque-spanish monolingual and bilingual infants at 3.5 months of age. 19(3):326–337.

[39] Clément Moulin-Frier, Sao M. Nguyen, and Pierre-Yves Oudeyer. Self-organization of early vocal development in infants and machines: the role of intrinsic motivation. *Frontiers in Psychology*, 4(1006), 2014.

[40] R Naatanen, A Lehtokoski, M Lennes, M Cheour, M Huotilainen, A Iivonen, M Vainio, P Alku, Rj Ilmoniemi, A Luuk, J Allik, J Sinkkonen, and K Alho. Language-specific phoneme representations revealed by electric and magnetic brain responses. *Nature*, 385(6615):432–434, 1997.

[41] Thierry Nazzi and Franck Ramus. Perception and acquisition of linguistic rhythm by infants. 41(1):233–243.

[42] Lw Norrix, E Plante, R Vance, and Ca Boliek. Auditory-visual integration for speech by children with and without specific language impairment. *Journal Of Speech Language And Hearing Research*, 50(6):1639–1651, 2007.

[43] D. Kimbrough Oller and Rebecca E. Eilers. The role of audition in infant babbling. *Child Development*, 59(2):441–449, 1988.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[45] Martha Pelaez, Javier Virués-Ortega, and Jacob L. Gewirtz. Contingent and noncontingent reinforcement with maternal vocal imitation and motherese speech: Effects on infant vocalizations. *European Journal of Behavior Analysis*, 12(1):277–287, 2011.

[46] Martha Pelaez, Javier Virués-Ortega, and Jacob L. Gewirtz. Contingent and noncontingent reinforcement with maternal vocal imitation and motherese speech: Effects on infant vocalizations. *European Journal of Behavior Analysis*, 12(1):277–287, 2011.

[47] Martin Pienkowski and Jos J. Eggermont. Cortical tonotopic map plasticity and behavior. *Neuroscience and Biobehavioral Reviews*, 35(10):2117–2128, 2011.

[48] Steven Pinker. The stuff of thought : language as a window into human nature, 2007.

[49] MA Pitt and AG Samuel. Lexical and sublexical feedback in auditory word recognition. *Cognitive Psychology*, 29(2):149–188, 1995.

[50] Robert Port. How are words stored in memory? beyond phones and phonemes. *New Ideas in Psychology*, 25(2):143–170, 2007.

[51] Friedemann Pulvermuller, Martina Huss, Ferath Kherif, Fermin Moscoso del Prado Martin, Olaf Hauk, and Yury Shtyrov. Motor cortex maps articulatory features of speech sounds.(neuroscience)(author abstract). *Proceedings of the National Academy of Sciences of the United States*, 103(20), 2006.

[52] F Ramus, MD Hauser, C Miller, D Morris, and J Mehler. Language discrimination by human newborns and by cotton-top tamarin monkeys. *Science*, 288(5464):349–351, 2000.

[53] Okko Räsänen. Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions. *Speech Communication*, 54(9):975–997, 2012.

[54] Nelli Salminen, Hannu Tiitinen, and Patrick May. Modeling the categorical perception of speech sounds: A step toward biological plausibility. *Cognitive, Affective and Behavioral Neuroscience*, 9(3):304–13, 2009.

[55] Kristina Simonyan, Hermann Ackermann, Edward F Chang, and Jeremy D Greenlee. New developments in understanding the complexity of human speech production. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 36(45), 2016.

[56] Kristina Simonyan and Stefan Fuertinger. Speech networks at rest and in action: interactions between functional brain networks controlling speech production. *Journal of neurophysiology*, 113(7), 2015.

[57] Wai Ting Siok, Zhen Jin, Paul Fletcher, and Li Hai Tan. Distinct brain regions associated with syllable and phoneme. *Human Brain Mapping*, 18(3):201–207, 2003.

[58] Steven W Smith. The scientist and engineer's guide to digital signal processing, 1997.

[59] Alena Stasenko, Cory Bonn, Alex Teghipco, Frank E. Garcea, Catherine Sweet, Mary Dombovy, Joyce Mcdonough, and Bradford Z. Mahon. A causal test of the motor theory of speech perception: a case of impaired speech production and spared speech perception. *Cognitive Neuropsychology*, 32(2):38–57, 2015.

[60] Joseph Paul Stemberger, Jeffrey Locke Elman, and Patricia Haden. Interference between phonemes during phoneme monitoring: Evidence for an interactive activation model of speech perception. *Journal of Experimental Psychology: Human Perception and Performance*, 11(4):475–489, 1985.

[61] Max Strange. Maxstrange/artieinfant: Thesis release, May 2019.

[62] Daniel Swingley. Contributions of infant word learning to language development. *Philosophical Transactions of the Royal Society B*, 364(1536):3617–3632, 2009.

[63] L. Ten Bosch, L. Boves, H. Van Hamme, and R.K. Moore. A computational model of language acquisition: The emergence of words. *Fundamenta Informaticae*, 90(3):229–249, 2009.

[64] Athena Vouloumanos and Janet F. Werker. Tuned to the signal: the privileged status of speech for young infants. *Developmental Science*, 7(3):270–276, 2004.

[65] Pei-Jung Wang, Ai-Wen Hwang, Hua-Fang Liao, Pau-Chung Chen, and Wu-Shiun

Hsieh. The stability of mastery motivation and its relationship with home environment in infants and toddlers. *Infant Behavior and Development*, 34(3):434–442, 2011.

[66] Anne S. Warlaumont, Megan K. Finnegan, and Tom Verguts. Learning to produce syllabic speech sounds via reward-modulated neural plasticity. *PLoS ONE*, 11(1), 2016.

[67] Anne S. Warlaumont, Gert Westermann, Eugene H. Buder, and D. Kimbrough Oller. Prespeech motor learning in a neural network using reinforcement. *Neural Networks*, 38:64–75, 2013.

[68] Janet F. Werker and Suzanne Curtin. Primir: A developmental framework of infant speech processing. *Language Learning and Development*, 1(2):197–234, 2005.

[69] Janet F. Werker, Linda Polka, and Judith E. Pegg. The conditioned head turn procedure as a method for testing infant speech perception. 6(3):171–178.

[70] Janet F Werker, H. Henny Yeung, and Katherine A Yoshida. How do infants become experts at native-speech perception? *Current Directions in Psychological Science*, 21(4):221–226, 2012.

[71] Gert Westermann and Nicolas Ruh. A neuroconstructivist model of past tense development and processing. *Psychological Review*, 119(3):649–667, 2012.

[72] LH Wurm and AG Samuel. Lexical inhibition and attentional allocation during speech perception: Evidence from phoneme monitoring. *Journal Of Memory And Language*, 36(2):165–187, 1997.

[73] Y. Yoshikawa, M. Asada, K. Hosoda, and J. Koga. A constructivist approach to infants' vowel acquisition through mother-infant interaction. *Connection Science*, 15(4):245 – 258, 2003.

[74] Ivan Yuen, Matthew H Davis, Marc Brysbaert, and Kathleen Rastle. Activation of

articulatory information in speech perception. *Proceedings of the National Academy of Sciences of the United States of America*, 107(2), 2010.