Neural network models for phonology and phonetics

Boersma, P.; Benders, T.; Seinhorst, K.

DOI
10.15398/jlm.v8i1.224

Publication date
2020

Document Version
Final published version

Published in
Journal of Language Modelling

License
CC BY

Link to publication

Citation for published version (APA):
Boersma, P., Benders, T., & Seinhorst, K. (2020). Neural network models for phonology and phonetics. *Journal of Language Modelling*, *8*(1), 103-177. https://doi.org/10.15398/jlm.v8i1.224

Neural network models for phonology and phonetics

Paul Boersma[1], *Titia Benders*[2], *and Klaas Seinhorst*[1]
[1] University of Amsterdam
[2] Macquarie University

ABSTRACT

This paper[1,2] argues that if phonological and phonetic phenomena found in language data and in experimental data all have to be accounted for within a single framework, then that framework will have to be based on neural networks. We introduce an artificial neural network model that can handle stochastic processing in production and comprehension. With the "inoutstar" learning algorithm, the model is

Keywords: phonology, neural networks, speech perception, historical linguistics

[1] Parts of this work were presented at the 31st Annual Meeting of the Deutsche Gesellschaft für Sprachwissenschaft, Osnabrück, March 2009; the 45th Annual Meeting of the Chicago Linguistic Society, April 2009; the KNAW Academy Colloquium on Language Acquisition and Optimality Theory, Amsterdam, July 2009; the 5th International Conference on Native and Non-native Accents of English, Łódź, December 2011; the 10th International Conference on Computational Processing of Portuguese Language in Coimbra, April 2012; the 20th Manchester Phonology Meeting, May 2012; the 2012 International Child Phonology Conference, Minneapolis, June 2012; the 19th Frysk Filologekongres, Ljouwert, June 2012; the 13th Conference on Laboratory Phonology, Stuttgart, July 2012; and the EGG Summerschool, Wrocław, August 2012. We thank the audiences, as well as Silke Hamann and Kateřina Chládková for their input. The research was sponsored by NWO grant 277-70-008 to Boersma, and NWO grant 021.002.095 to Benders.

[2] The learning algorithm developed in this manuscript has subsequently been used and/or elaborated by Benders (2013); Chládková (2014); ter Schure *et al.* (2016); and Seinhorst *et al.* (2019).

able to handle two seemingly disparate phenomena at the same time: gradual category creation and auditory dispersion. As a result, two aspects of the transmission of language from one generation to the next are integrated in a single model. The model therefore addresses the hitherto unsolved problem of how symbolic-looking discrete language behaviour can emerge in the child from gradient input data from her language environment. We conclude that neural network models, besides being more biologically plausible than other frameworks, hold a promise for fruitful theorizing in an area of linguistics that traditionally assumes both continuous and discrete levels of representation.

1 WHY A COMPREHENSIVE MODEL MUST BE BASED ON NEURAL NETWORKS

What will be the ultimate model of phonology and phonetics and their interactions? It will have to be a model that accounts for at least four types of valid behavioural data, namely 1) the generalizations that phonologists have found within and across languages, 2) the phenomena that psycholinguists and speech researchers have found by observing speakers, listeners, and language-acquiring children, 3) the mergers, splits, chain shifts and other sound change phenomena found by historical phonologists and dialectologists, and 4) the phenomena that have been observed when languages come in contact, such as loanword adaptations. Besides having to account for all these types of behavioural data, the model will have to be compatible with what is known about the biology of the human brain, because that is where language is produced and comprehended. In this paper we argue that the ultimate model has to be reductionist, i.e. that it has to consist of artificial neural networks. We provide a first proposal of a neural network model that can handle two important aspects of the transmission of a sound system from one generation to the next, namely category creation and auditory dispersion, and we simulate the model on a range of synthetic data.

1.1 A model of phonological and phonetic representations and knowledge

If the model contains levels of representation, it may look like Figure 1, which can be thought of as containing the minimum number of levels needed for a sensible description: phonetics seems to require at least an Auditory Form (AudF, specifying a continuous stream of sound) and an Articulatory Form (ArtF, specifying muscle activities), and phonology seems to require at least an Underlying Form (UF, containing at least lexically contrastive material) and a Surface Form (SF, containing a whole utterance divided up in prosodic structure such as syllables); the Morpheme level connects the phonology to the syntax and the semantics in the lexicon.

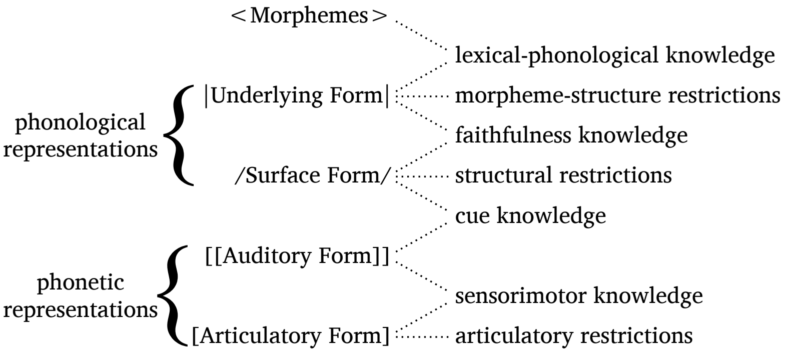


Figure 1: Levels of representation and stored knowledge in a model of phonology and phonetics

The five levels in Figure 1 are a simplified combination of what phonologists have been proposing in models of phonological production (e.g. van Wijk 1936: 323; Trubetzkoy 1939; Kiparsky 1982) and what psycholinguists have been proposing in models of comprehension (e.g. McClelland and Elman 1986; Cutler *et al.* 1987) and production (e.g. Levelt *et al.* 1999). These specific five levels, and the special way in which they are connected in Figure 1, were proposed by Boersma (1998, 2007) and Apoussidou (2007). In numerous papers, Boersma and co-workers have investigated the capability of this "Bidirectional Phonology and Phonetics" (BiPhon) model to account for experimental as well as linguistic data (for an overview, see Boersma 2011). The model has hitherto used the decision mechanism of Optimality Theory (OT) and can therefore be called BiPhon-OT.

The present paper introduces the neural-network (NN) edition of the model, which we call BiPhon-NN.

Language users have knowledge of the relationships between levels of representation. In Figure 1, such relationships exist between adjacent levels only, so that the language user has knowledge about sensorimotor, cue, faithfulness (phonological) and lexical relationships. The language user also has knowledge about restrictions within levels: the articulatory, structural and morpheme-structure restrictions. In OT, all this knowledge is represented as a grammar consisting of ranked constraints; in NN models, this knowledge is represented as a long-term memory consisting of connection weights.

1.2 *Phonological and phonetic processes*

A comprehensive model has to take into account the behaviour of the speaker, the listener, and the learner. Figure 2 shows the various *processes* that can be distinguished when travelling the levels of representation of Figure 1. Globally, the path from AudF to Morphemes following the upward arrows in Figure 2 is *comprehension*, i.e. the task of the listener, and the path from Morphemes to ArtF following the downward arrows is *production*, the task of the speaker. More locally, there are partial processes. The local mapping from UF to SF is *phonological production*, an example being the mapping from an underlying two-word sequence |an#pa| ("#" denotes a word boundary) to the phonological surface structure /.am.pa./ ("." denotes a syllable boundary) in a language with nasal place assimilation. At the interface between phonetics and phonology, the local mapping from AudF to SF is (prelexical) *perception*, an example being the mapping from concrete continuous formant values to abstract discrete vowel categories.

The partial processes and their acquisition have been modelled in various frameworks. Phonologists have been modelling phonological production within OT since Prince and Smolensky (1993/2004), and its acquisition since Tesar and Smolensky (1998). Word recognition was modelled with neural networks by Norris (1994) in the Shortlist model, and prelexical perception was modelled with neural networks by Weenink (2006) and within BiPhon-OT by Boersma (1997) and Escudero and Boersma (2004). The present paper in Section 5 models

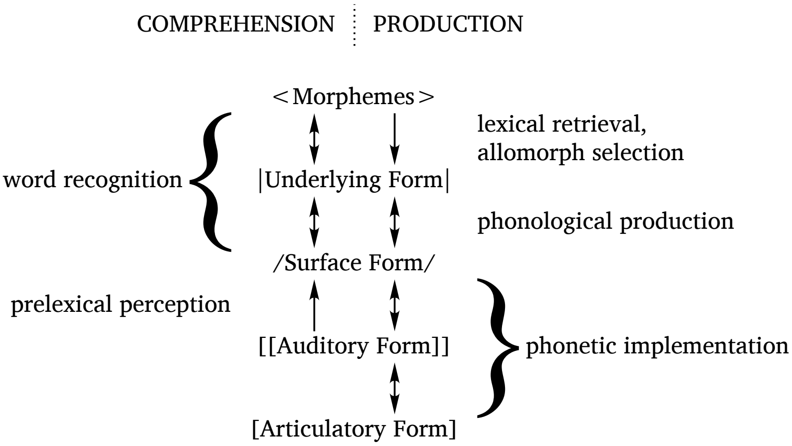


Figure 2: Processes in a comprehensive model of phonology and phonetics

the development of *category creation* in the AudF-to-SF mapping. The emergence of an early stage of category creation, namely the perceptual magnet effect (which was observed in the lab by Kuhl 1991), has been modelled before with neural networks by Guenther and Gjaja (1996) and with BiPhon-OT by Boersma *et al.* (2003).

The way in which the language user's knowledge is represented in Figure 1 suggests that the same knowledge is used for both directions of processing in Figure 2, i.e. for comprehending and producing speech. Within OT, this *bidirectionality* was first argued for by Smolensky (1996). Specifically, it has often been argued that the same structural constraints play a role in comprehension as well as in production (Tesar 1997; Tesar and Smolensky 1998, 2000; Boersma 1998, 2000, 2007, 2009; Pater 2004), sometimes with very dissimilar effects (Boersma and Hamann 2009). For the present paper it is relevant that the "cue knowledge" at the interface of phonology and phonetics is bidirectional, i.e. used in both prelexical perception and phonetic implementation (Boersma 2009): the same knowledge that allows one to perceive a loud high-frequency noise as /s/ forces one to implement the surface phoneme /s/ as a sound with a loud high-frequency noise. In Section 6 we model within BiPhon-NN the acquisition of *auditory dispersion*, i.e. the evolution of optimal distances at AudF between the members of phoneme inventories at SF. This acquisition has been modelled before within exemplar theory by Wedel (2004, 140–169; 2006, 261–269) and in BiPhon-OT by Boersma and Hamann (2008); in both cases, bidirectionality was a crucial element of the explanation, as explained in detail in Section 6.

Thus, the perceptual magnet effect and auditory dispersion have both been modelled before, although rarely within the same framework (with BiPhon-OT as a possible exception).

1.3 *The need to model it all at the same time*

There are at least two reasons why one would want to model all the processes of Section 1.2 within a single comprehensive model. One reason is that there are phenomena whose complete explanation necessarily requires all levels of representation, and the other reason is that there seem to exist processes that require an interaction between levels that are far away from each other in Figure 1 or 2. We discuss these reasons now, with the goal of finding candidate comprehensive modelling frameworks.

1.3.1 Comprehensive processes

There exist seemingly unitary processes whose explanation nevertheless requires all levels of representation. One such process is loanword adaptation, where the input (the foreign stream of sound that impinges on the borrower's ear) and the output (the borrower's phonetic production) are the only direct observables. If one wants to understand this phenomenon solely on the basis of acquired L1 behaviour, one has to assume that the borrower starts by filtering the incoming auditory form through L1-specific cue knowledge and L1-specific structural constraints into a phonological surface structure (see Figures 1 and 2), then stores it as a new morpheme in the lexicon with an appropriate underlying form. When speaking, the borrower takes this morpheme and underlying form, filters the latter with her L1-specific phonological knowledge, then filters the result again with her phonetic implementation device, which computes an auditory form and an articulatory form, perhaps filtered by L1-specific articulatory restrictions. An explanation of loanword adaptation, therefore, requires all arrows in Figure 2, as has been argued in detail by Boersma and Hamann (2009). Another phenomenon whose explanation requires all levels of representation is first-language acquisition. This happens much more slowly than the initial adaptation of a loanword, but is also much more central to linguistic theory and experimentation. The search we have to embark on, therefore, is for a single comprehensive framework.

Distant interactions 1.3.2

The arrows in Figure 2 only connect levels that are adjacent. Thus, an incoming sound at AudF first activates a representation at SF, which then activates a representation at UF, which then activates one or more morphemes at the topmost level; there are no more direct routes that skip a level.

However, there is evidence that the partial processes are not entirely sequential. Feedback from "later" to "earlier" levels of representation has been identified experimentally and theoretically in several locations of processing, and several models that exhibit such interactions have already been proposed. In comprehension, lexical influence (from the Morpheme level) back to prelexical perception (AudF-to-SF) was attested by Ganong III (1980), who found that an auditory sequence that is ambiguous between /dæ/ or /tæ/ (for English listeners) is perceived as /dæ/ if followed by [ʃ] and as /tæ/ if followed by [sk], simply because |dæʃ| and |tæsk| correspond to English words, while |tæʃ| and |dæsk| do not; this effect was modelled with neural networks by McClelland and Elman (1986) and with BiPhon-OT by Boersma (2009, 2011). Likewise, semantic considerations above the Morpheme influence the access of underlying forms in the mapping from SF to UF (Warren and Warren 1970). In production, allomorph selection at UF or higher is sometimes determined by "later" considerations at SF: the choice between |vjø| and |vjɛj| 'old-MASC' in French is determined by whether the next word happens to start with a consonantal segment or not, as modelled with BiPhon-OT by Boersma and van Leussen (2017). Likewise, phonetic considerations such as articulatory effort (at ArtF) and cue quality (between SF and AudF) may influence choices in the phonology (between UF and SF), as modelled by Boersma (1998, 2007). Also, cue knowledge and articulatory constraints must interact with each other in the phonetic implementation process.

As a result of these examples of *interactive* processing, most of the arrows in Figure 2 are two-sided. Levels that are activated "later" in comprehension or production can thereby influence "earlier" levels backwards. In NN models, interactivity is implemented by having activity spread bidirectionally (McClelland and Elman 1986); in BiPhon-OT the interactivity is implemented by having candidates be entire paths from AudF to Morpheme in comprehension or from Morpheme to ArtF in production (Boersma 2007, 2009, 2011; Apoussidou 2007; Berent *et al.* 2009).

The existence of such feedback in processing is controversial in some locations (Norris *et al.* 2000 deny the influence of the lexicon on prelexical perception, and Hale and Reiss 2000 deny any influence of phonetic considerations on phonological production). For the time being, however, we assume interactivity is everywhere. The need for a comprehensive model does not depend on whether such interactivity is only apparent or is an integral element of the underlying mechanism.

1.4 *Choosing the framework that models it all: neural networks*

When discussing existing models in Section 1.1 through Section 1.3, we identified three frameworks: neural networks, exemplar theory, and OT.

At first sight, BiPhon-OT might seem to be the best framework, because it provided an account of all of the processes mentioned. However, this is deceptive, because it did not provide an account of all the processes *combined*. When modelling phonological category creation (Boersma 1998: ch.8; Boersma *et al.* 2003), the BiPhon model shares with NN category creation models (Guenther and Gjaja 1996) the assumption that phonological categories emerge from the distributions of auditory forms in the child's environment. Both computational models successfully arrive at a stage of continuous perceptual warping (an incoming sound is received as a slightly different sound because of distributional learning), but linguistic modelling in OT has to stop there, because it has to assume that categories are discrete. This discrepancy between the gradience (continuity) of category creation that is needed in an emergentist model, and the discreteness of categories that is needed to do OT phonology entails the failure of OT as a comprehensive framework for emergentist phonology and phonetics. Moreover, OT's biological plausibility is low, because it works with nearly infinite lists of candidates, which is especially problematic if we have five levels of representation; typically, the number of candidate paths to evaluate is exponential in the length of the input (both in comprehension and in production) as well as exponential in the number of levels of representation.

Superficially, exemplar theory (Goldinger 1996) might be expected to do better with respect to a transition from continuous to discrete, because this theory can at least be seen to handle the reverse transition when massive storage of single discrete events leads to observed continuous knowledge. However, despite its long existence, the theory has not yet been able to model even the most straightforward of phonological processes, such as productive nasal place assimilation (Boersma 2012). More crucially, work specifically addressing the acquisition of categories (Kruschke 1992; Pierrehumbert 2001) presupposes pre-existing category labels, i.e. it models the emergence of the link between categories and sound but not the emergence of the category labels themselves.

This leaves neural network modelling as the only option. If Figure 1 is implemented in a neural network, each of the five levels of representation should be thought of as a large set of network *nodes*, each of which can be active or inactive (or, in a time-smoothed view, more active or less active). The pattern of activity of these nodes forms the current representation at that level. The processes of Figure 2 can be regarded as the spreading of activity between and within levels; the knowledge in Figure 1 is stored as connection weights, i.e. the strengths of the connections between the nodes. We show in Section 5 that if the elements of representations are distributed over multiple nodes, they can start out as continuous and gradually come to exhibit more discrete behaviour during acquisition, thus ensuring the compatibility between underlying continuity and observed discreteness. One and the same framework, then, succeeds in accounting for both symbolic and subsymbolic behaviour. As far as biological plausibility goes, neural networks form the best of the three frameworks as well: the number of connections in a NN model tends to rise linearly with the number of levels of representation, and linearly or quadratically with the size of the representations.

We confess here that we choose NN modelling not only because it wins out by elimination, but also because it is reductionist: in the end, it is uncontroversial that humans represent language in neural networks in their brains, and both OT and exemplar theory work at a higher level of abstraction. If the abstractions fail, one has to go one level of concreteness deeper.

Artificial neural networks differ in their structure, in their activation spreading rules, and in their learning rules. To assess the appropriateness of various neural networks for bidirectional phonology and phonetics, Sections 2 through 4 start by looking at a traditional toy example of phonological production, and then establish what common elements of artificial neural networks are needed or unnecessary and why. Readers who want to skip these justifications and are also thoroughly familiar with neural net modelling can jump ahead to the conclusion and summary in Section 4.8. Sections 5 and 6 then show that with these elements we can build a shallow network that can create categories (Section 5) and exhibits auditory dispersion (Section 6). Let's proceed to looking at the ingredients of our linguistic NN model.

2 NODES, CONNECTIONS, WEIGHTS AND ACTIVITIES

The neural network type of our choice should at least be able to share the properties that made BiPhon-OT successful in modelling language phenomena: stochasticity (it should replicate environmental probabilities) and bidirectionality (it should work both top-down and bottom-up). This section shows that these two desirable properties can be achieved in a network architecture with probabilistic nodes and bidirectional connections. For initial simplification, we work with "local" representations in this section, because these allow us to investigate (in Section 4) the theoretical asymptotic behaviour of our networks, i.e. to investigate what kinds of general cognitive problem our networks must be able to solve after learning; our real proposal in later sections has "distributed" representations instead, for reasons we make clear in Section 5.

A toy example: phonological production 2.1

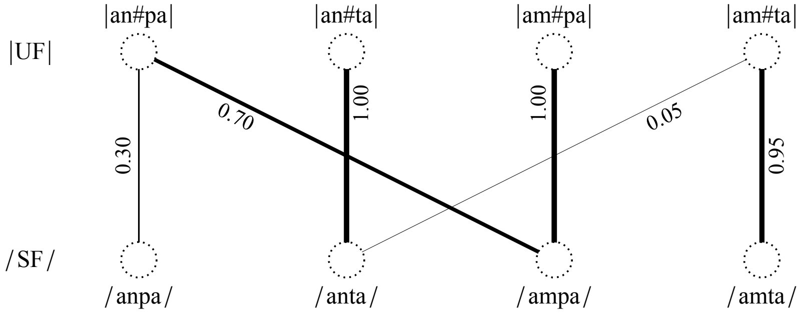
Following Figure 2, phonological production, viewed in isolation, is the mapping from Underlying Form (UF) to Surface Form (SF). Using terms that are familiar from both the neural network literature (Rosenblatt 1958) and OT (Prince and Smolensky 1993/2004, Section 1.1), the Underlying Form is the *input* of this mapping and the Surface Form is the *output*. For simplification, we start with a toy language that models only the relationship between UF and SF, although we do so in both directions of processing.

Our toy language has only four possible underlying utterances, each of which consists of two words. The first word is either underlyingly |an| or |am|, and the second word is either |pa| or |ta|. The four underlying utterances are therefore |an#pa|, |an#ta|, |am#pa| and |am#ta|, where "#" stands for the word boundary. In the surface form, the language exhibits nasal place assimilation in a manner reminiscent of Dutch: an underlying coronal nasal tends to assimilate to the place of any following consonant, so that underlying |an#pa| becomes /ampa/ on the surface; meanwhile, an underlying labial nasal tends not to assimilate: |am#ta| becomes /amta/. As in real languages, the tendencies are not true 100% of the time: the assimilation of the coronal nasal is optional, and likewise, the labial nasal does assimilate in a small minority of cases. For our example we suppose that underlying |an#pa| becomes assimilated /ampa/ on the surface 70% of the time, but the "faithful" form /anpa/ 30% of the time, and that underlying |am#ta| becomes faithful /amta/ 95% of the time, and assimilated /anta/ 5% of the time.

This probabilistic state of affairs is a situation that (Stochastic) OT is known to be able to represent (e.g. Boersma and Hayes 2001), because an existing learning algorithm for Stochastic OT (the "GLA") typically turns a learner into a probability matcher. In comprehension, an auditory form that was intended by the speaker as the surface form /A/ in 70% of the cases and as the surface form /B/ in 30% of the cases, will come to be perceived by the GLA perception learner as /A/ in 70% of the cases and as /B/ in 30% of the cases (Boersma 1997). In production, an underlying form that is produced in the learner's language environment as the surface form /C/ in 70% of the cases and as the surface form /D/ in 30% of the cases will come to be produced by the GLA production learner as /C/ in 70% of the cases and as /D/ in 30% of the cases (Boersma and Hayes 2001). Our NN model should be able to replicate this or a similar kind of optimal behaviour.

There are several ways to represent this toy language in a neural network. The most straightforward and OT-like (and probably least realistic) way is to represent each possible underlying utterance (input) with one *node*, and each possible output utterance as one node. This is done in Figure 3, where each of the four possible underlying forms shows up as a single node along the top and each of the four surface candidates shows up as a single node along the bottom.

Figure 3: A network that performs phonological production

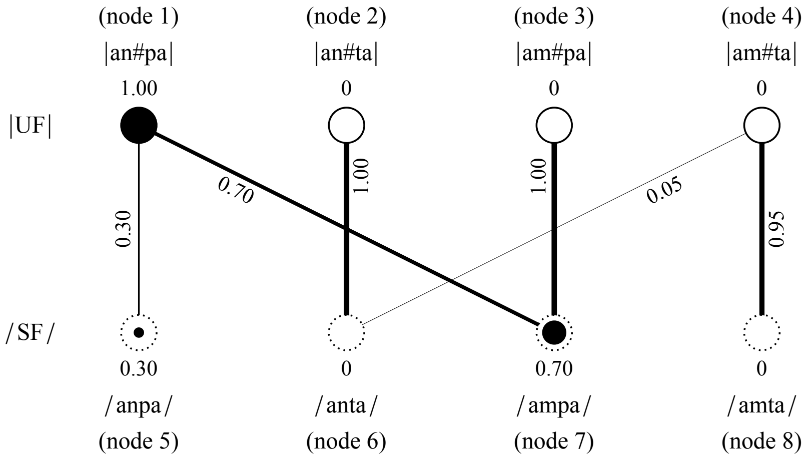


Biologically, a node can be regarded as representing a neuron (or small group of neurons) in the cerebral cortex. Representing an entire linguistic form with a single node (a *local* representation), as we do here, is an unrealistic oversimplification, employed here only for purposes of illustration; more realistic *distributed* representations, where a single phonological category is represented by multiple nodes, appear in Section 5.

In Figure 3, each node is visualized as a dotted circle. Each of the four UF nodes is *connected* to each of the four SF nodes, although only six of the 16 connections are visible. Biologically, a connection corresponds to a synapse (point of contact) between an outgoing branch of one neuron and a receiving branch of another neuron. Such a synapse is unidirectional: it permits an electric signal to flow from one neuron to another. In general, therefore, the total strength of the synapses that carry signals from neuron A to neuron B is not equal to the total strength of the synapses that carry signals from neuron B to neuron A. Nevertheless, we maintain in this paper the simplification that the strength of the connection from node A to node B equals the strength of the connection from node B to node A, and that it can therefore be called the strength of *the* connection *between* nodes A and B. Such bidirectional connections are known to provide stability in neural network models (Hopfield 1982; O'Reilly 1996), and they guarantee the bidirectionality (Section 1.2) of the BiPhon model, thus providing the desired dispersion effect in Section 6. The present paper can do with, and indeed crucially employs, bidirectional connections; if in future modelling this simplification turns out to be untenable, bidirectionality should then be dispensed with.

In NN modelling, connection strengths are called *weights*. The weight of the connection between the input node |an#pa| and the output node /anpa/ is 0.30, and this is visualized in Figure 3 in two ways: the number 0.30 is written next to this line, and the thickness of the connection line is 0.30. Biologically, the connection weight indeed corresponds to the thickness of the synapse, i.e. the area with which the sending neuron is connected to the receiving neuron. When a biological neuron fires, a neuron with which it has a thick (strong) synapse will be influenced more strongly than a neuron with which it has a thinner (weaker) synapse (our simplified artificial neurons do not actually fire; see Sections 2.2–2.5). In the figure, therefore, thicker lines denote stronger information flows than thinner lines. For instance, the weight of the connection between |an#pa| and /ampa/ is 0.70, which is stronger than that between |an#pa| and /anpa/, because the underlying form |an#pa| should send stronger signals to /ampa/ than to /anpa/ in this toy language. Likewise, the weight of the connection between |an#pa| and /anta/ is zero, because we never want |an#pa| to be realized as /anta/; this zero-weight connection is not visible in the figure (the line has zero width). Also, an underlying "homorganic" |an#ta| is always realized as /anta/, and this is reflected with the number 1.00 next to the relevant connection line in the figure. We will show that with these chosen connection weights, the network in Figure 3 can indeed simulate the data of the toy language if the network has four common additional properties: all-or-none activation of the input nodes (Section 2.2), additive excitation of the output nodes (Section 2.3), a linear

Figure 4: The production of underlying |an#pa|



excitation-to-activity function (Section 2.4), and a linear activity-to-probability function (Section 2.5). We illustrate these concepts with Figure 4, which shows the production of underlying |an#pa|.

2.2 *All-or-none activation of the input nodes*

To compute how the network handles an incoming underlying form, we apply an *activity pattern* to UF and compute from it the activity pattern that will arise at SF. To see what the network does to an underlying |an#pa|, we activate the |an#pa| node by setting its activity to 1.00. This is shown in two ways in Figure 4: by painting the whole node in black, and by drawing the number 1.00 above the node. At the same time, we set the activities of the three remaining underlying forms to 0, which is indicated in the figure by not painting these three nodes.

Biologically, an activity can be thought of as a firing rate. A node with an activity of 1.00 can be seen as a neuron (or group of neurons) with a maximum firing frequency of, say, 10 spikes per second (Buzsáki and Mizuseki 2014); a node with an activity of 0 can be seen as a neuron (or group of neurons) with a minimum firing frequency (say, 0.1 spikes per second). In this paper we ignore the separate spikes and employ only continuous activities, usually between 0 and 1 (see Section 2.5).

The circles for the UF nodes in Figure 4 look different from those for the SF nodes. In the phonological production process, the UF level is the input, so that the activities of the four UF nodes will be held constant during evaluation. In neural-network terminology, the UF nodes are *clamped* (kept fixed). This is indicated in the figure by the circles for the UF nodes now having solid rather than dotted edges. By contrast, the SF level is the output of the process, so that the activities of the four SF nodes must be free to adapt themselves to the activities of the input nodes; dotted circles in the figure visualize the fact that the output nodes are *unclamped.*

Additive excitation of the output nodes 2.3

When an input node is activated, as node |an#pa| is in Figure 4, the information about its activity will spread toward the nodes with which it is connected: the activity will *excite* every connected node to some extent. For instance, in Figure 4, node |an#pa| has activity 1.00 and the connection between |an#pa| and /ampa/ has weight 0.70. The amount to which |an#pa| will excite /ampa/ is the product of the input activity and the connection weight, i.e. $1.00 \cdot 0.70 = 0.70$. Likewise, node |am#pa| has activity 0 and the connection between |am#pa| and /ampa/ has weight 1.00; |am#pa| will therefore excite /ampa/ by an amount $0 \cdot 1.00 = 0$. Node |an#ta| excites /ampa/ by an amount 0 (the activity of |an#ta|) times 0 (the weight of the connection from |an#ta| to /ampa/), which is $0 \cdot 0 = 0$, and so does |am#ta|.

Biologically, these four excitations can be regarded as "postsynaptic potentials", rises in the potential (in millivolts) of the membrane of the receiving neuron. These rises tend to be *additive*, i.e. all the small excitations add up to yield the total excitation of the receiving neuron (Lorente de Nó 1938). Artificial neural network models also tend to assume additive excitation. Thus, the total excitation of /ampa/ becomes $0.70 + 0 + 0 + 0 = 0.70$. In a formula, the excitation of the output nodes, i.e. nodes 5 through 8, can be computed as

(1) $$e_j = \sum_{i=1}^{4} w_{ij} a_i \qquad (\text{for } j = 5..8)$$

where a_i is the activity of UF node i, and w_{ij} is the weight of the connection between UF node i and SF node j.

2.4 *Activity of the output nodes*

When a node is excited, it becomes active itself. Biologically, this corresponds to the fact that if the membrane potential of a neuron rises, the probability that it will fire increases; in a continuous (and simplified) view of neuronal activity (Perkel and Bullock 1969) this means that if the time-averaged membrane potential rises, the firing frequency of the neuron will rise as well. The simplest assumption about the relation between excitation and activity is that it is *linear*, i.e. the activity rises and falls with the excitation by a constant factor. If this factor is 1, the activity of an SF node in our example becomes equal to its excitation:

(2) $a_j = e_j \qquad (\text{for } j = 5..8)$

With this *identity activation function*, activating |an#pa| causes an activity of 0.70 in node /ampa/. This number is written over the node in the figure and is also visible as the size of the black disk in that node. Likewise, activating |an#pa| causes an activity of 0.30 in node 5, which is visualized in the figure as the small black disk in that node.

Other excitation-to-activity functions are possible. If one wants to make sure that the activities of the SF nodes do not become negative, (which seems reasonable, given the biological interpretation of the activity as a firing frequency), one could simply clip the activity from below, maintaining linearity of all activities above 0:

(3) $a_j = \max\left(0, e_j\right) \qquad (\text{for } j = 5..8)$

For our toy example, this *rectifying activation function* (Hahnloser *et al.* 2000) works in the same way as the identity activation function of (2), because all excitations are non-negative; in Sections 5 and 6, however, the clipping will be crucial (see Section 5.9.5 for details). Finally, if one wants to take into account that biological firing frequencies have not just a minimum but also a maximum, one could apply a "top-sigmoid" clipping, which is linear for small excitations and goes to 1 smoothly for large excitations:

$$(4) \qquad a_j = \max\left(0, \frac{2}{1+e^{-2e_j}} - 1\right) \qquad (\text{for } j = 5..8)$$

For our toy example, combining the assumption of additive excitation (the contributions from the four underlying forms are added up) and the assumption of the identity excitation-to-activity function (the activity of an output node equals its excitation) causes the activity of an SF node to become the sum of the activities from the input nodes, weighted by the weights of the connections.

Probabilistic interpretation of the activity of the output nodes 2.5

Having computed the activities of the output nodes is not the end of the story. If we want to use neural networks to model linguistic behaviour, we will have to provide a behavioural interpretation of the result in Figure 4. After all, there is no third level of representation that the activities on nodes 5 through 8 could feed into (in this toy example). The only behaviour one can then think of is that the virtual speaker chooses one of the four surface forms to actually produce. The question is: which SF will the virtual speaker choose?

One possible answer is that the speaker chooses the node that has the highest activity, i.e. the node /ampa/. This is an option often found in neural network modelling, especially in competitive learning (Grossberg 1976, 1987; Rumelhart and Zipser 1985). Here, however, this option would throw away the /anpa/ candidate entirely, and such nonstochastic behaviour is not desirable if we want to model the 70–30 variation of our toy language.

Another possible answer is that the speaker somehow produces both /ampa/ and /anpa/. Such a mix might be imaginable at a continuous level of representation such as ArtF, where we can imagine what mixed gestures would look like, but the notion of mixed phonological representations at SF is difficult to envision (but see Section 5.6).

The third possible answer is that the activities denote probabilities: /ampa/, with an activity of 0.70, is chosen with a probability of 70%, and the only other competing candidate /anpa/, which has an activity of 0.30, is chosen with a probability of 30%. This means that if we ask the network to produce an SF from the input |an#pa| 1000 times, the network will say "/ampa/" approximately 700 times, and "/anpa/" approximately 300 times. In general, then, the probability of output candidate j is its activity, scaled by the sum of all output activities:

$$(5) \qquad P_j = \frac{a_j}{\sum_{k=5}^{8} a_k} \qquad (\text{for } j = 5..8)$$

Thus, since the candidate /ampa/ has an activity of 0.70 and the other candidates have activities of 0.30, 0, and 0, the probability of /ampa/ can be computed under the linear activity-to-probability assumption of (5) as $0.70/(0.30 + 0.70 + 0 + 0) = 70\%$. Equation (5) is known in psychology as *Luce's choice axiom* (Luce 1959: 23), and it can apply to any type of non-negative numbers a_j that represent strengths (or weights, or activations, or saliences) of the candidates j.

Such an interpretation of an activity as a relative probability has a biological correlate. If activity can be regarded as firing frequency, and /ampa/'s activity is 0.70 while /anpa/'s activity is 0.30, then node /ampa/ fires 2.333 times as often as node /anpa/ in any given period of time. This means that if, from a certain moment in time on, one waits until either node /ampa/ or node /anpa/ fires, the odds will be 7 to 3 that node /ampa/ fires earlier than node /anpa/. In other words, there will be a probability of 70% that node /ampa/ fires first, and a probability of 30% that node /anpa/ fires first. If the first node to fire determines the speaker's behaviour, the relative activities have apparently determined the relative probabilities of the behaviour.

Different interpretations of the relation between activity and probability are nevertheless possible. In the *Boltzmann machine* (Ackley *et al.* 1985), the probabilities are

$$(6) \qquad P_j = \frac{e^{a_j/T}}{\sum_{k=5}^{8} e^{a_k/T}} \qquad (\text{for } j = 5..8)$$

where T is called the *temperature*. Equation (6), known in modern machine learning as *softmax*, is due to Boltzmann (1868), is a special case of Luce's choice axiom, and can apply to any type of numbers a_j (even negative ones) that represent strengths of the candidates j. The simpler linear relation of (5), however, suffices for the present paper, because we work solely with non-negative activities (see especially Section 5.6).

Bidirectionality violated? 2.6

The network of Figure 3 works correctly in the production direction, i.e. with UF as the input and SF as the output. In the spirit of the BiPhon model we would like it to work equally well in the comprehension direction, i.e. with SF as the input and UF as the output. To model the recognition of an incoming /ampa/ as an underlying sequence of words, we can start by clamping the four SF nodes by keeping the /ampa/ node at a constant activity of 1.00 and the other three nodes constantly at zero. According to Figure 3 and the procedure of (1) and (2), the underlying form |an#pa| will get an activity of 0.70 and the underlying form |am#pa| will get an activity of 1.00. Apparently, the network prefers |am#pa| over |an#pa| when it listens.

This situation is fine if the underlying forms |an#pa| and |am#pa| occur equally often in the language environment: the network's preference then mimics the likelihood with which each of the two underlying forms was intended, given the surface form /ampa/. If, however, coronals occur in word-final position three times more often than labials do (which is approximately true for Dutch and English), the underlying form |an#pa| is three times more likely a priori than |am#pa| is. According to Bayes (Laplace 1812), this should shift the preference of a listener toward |an#pa|, but in the network of Figure 3 this is not taken into account. In fact, the weights are conditional probabilities on UF only, not on SF.

This asymmetry between comprehension and production is a general property of symmetric connections. It cannot be completely solved, but it can be made equally (un)problematic for both directions of processing, as we do in Section 4.

Section 2 has shown that an artificial neural network can replicate the decision mechanism of (Stochastic) OT or (Noisy) HG; in other words, the network mimics the decision mechanism of a probabilistic grammar. It is unsatisfying, though, that each full utterance is represented as a single node. In a more realistic network, the representation of each phonological element will be *distributed* over multiple nodes. Such a network is discussed in Section 5. Understanding such a network, however, requires understanding how the activities of equation (1) come about in processing (Section 3), and how the weights in Figure 3 come about in learning (Section 4).

3 ACTIVITY SPREADING

In the example of Section 2, the initially unknown activities of the unclamped (output) nodes could be computed directly by equations (1) and (2) from the given activities of the clamped (input) nodes. Such a direct computation is possible for simple two-level mappings as in that example, but with larger networks, in which information flows bottom-up, top-down and within levels simultaneously, a direct computation is no longer possible, because the activities of some unclamped nodes come to depend on the activities of other unclamped nodes that themselves are not known from the start.

The general solution is to compute the activity in the unclamped nodes iteratively, i.e. in small steps, from the given activities of the clamped nodes, and let the network gradually approach its equilibrium, i.e. a final state in which its activities stop changing. Such gradual activity spreading bears similarities with how activity spreads through biological neural networks, and proceeds as follows. After applying some known activities to the clamped nodes, we let the excitations (and activities) of the unclamped nodes start at zero, and we then update these excitations in small steps several hundreds of times. In the example of Section 2, the excitation in the output nodes 5 through 8 starts at zero, and is incremented at every time step (say, every millisecond) by an amount Δe_j given by

(7) $$\Delta e_j = 0.01 \cdot \left(\sum_{i=1}^{4} w_{ij} a_i - e_j \right) \qquad (\text{for } j = 5..8)$$

where the factor of 0.01 is the *spreading rate*.

For our specific toy example, it is easy to show that the general equation (7) indeed produces the end result of equation (1) after some time. Consider the situation for the output node /ampa/ at time 0. We already know that $\Sigma_{i=1}^{4} w_{i7} a_i = 0.70$, so at time zero, when $e_7 = 0$, Δe_7 will be $0.01 \cdot (0.70 - 0) = 0.007$. Therefore, e_7 becomes 0 (its previous value) plus 0.007 (the increment), which makes 0.007. At the next time step, $\Sigma_{i=1}^{4} w_{i7} a_i$ is still 0.70, but e_7 is 0.007, so that the increment Δe_7 is $0.01 \cdot (0.70 - 0.007) = 0.00693$, just 1% smaller than the previous increment. As a result, the new value of e_7 becomes $0.007 + 0.00693 = 0.01393$. Figure 5 shows what happens if this procedure is repeated 500 times (i.e. for, say, half a second): while the increment decreases exponentially by a factor of 0.99 at each time step, the excitation (and therefore the activity) of output node 7 grows asymptotically toward 0.70.

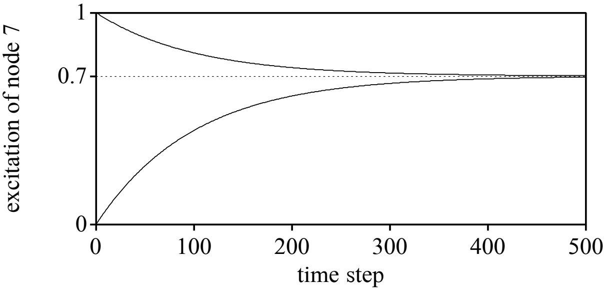


Figure 5:
The time path of the excitation (and activity) of node /ampa/.
Bottom curve: starting from 0.
Top curve: starting from 1.00

One could have predicted the end state of our toy example directly from (7), by realizing that in the equilibrium situation Δe_7 goes to zero. Equation (7) tells us that in that case $\Sigma_{i=1}^{4} w_{i7} a_i - e_7$ must go to zero as well. This means that e_7 goes to $\Sigma_{i=1}^{4} w_{i7} a_i$, i.e. to 0.70, so the activity, by (2), also goes to 0.70, which is the activity in Figure 4. This also shows that the starting value of the excitation does not matter: the excitation will go to 0.70 no matter where it started; as an illustration, Figure 5 also shows how the excitation develops if it starts at 1.00. This kind of reasoning from zero increments is a general trick to predict what the final situation will look like, given a formula for increments.

The evolution of the activities toward a constant final state, as in Figure 5, is general for symmetric networks (Hopfield 1982; Ackley *et al.* 1985). After enough time, each node j reaches a stable equilibrium state where its excitation stops changing, i.e. where Δe_j approaches zero. As a result, the whole network reaches equilibrium, i.e. the excitations of all its nodes stop changing. Symmetric networks, where w_{ij} equals w_{ji}, are guaranteed to move toward such a stable final state.

The general formula for the activity spreading toward an unclamped node j from its (clamped or unclamped) neighbours i is

$$(8) \qquad \Delta e_j = \eta_a \left(\sum_{\text{connected nodes } i} (w_{ij} - \textit{shunting}\ e_j)\, a_i - \textit{excitationLeak}\ e_j \right)$$

Here, η_a is the spreading rate, which in our simulations is kept constant at a value of 0.01. The *excitation leak* factor was set to 1 in (7), but could be set to higher values if we want to reduce the ultimate activity values. The *shunting* factor (Grossberg 1976) is included here only for completeness; it is set to 0 in all simulations in this paper.

4 A LEARNING RULE FOR BIDIRECTIONAL LINGUISTICS: INOUTSTAR

The representations and processes discussed in Sections 2–3 are transient things: they come and go every few seconds as the listener receives more speech or the speaker produces more speech. The connection weights contain more persistent information, namely the aspects of knowledge seen in Figure 1. These weights can *learn* from experience: they change only slowly over the months and years as the child is acquiring her language. In this section we identify a learning rule for our stochastic bidirectional artificial networks: we show that out of a family of Hebbian-like learning rules the only rule that meets the requirements of stochasticity and symmetric bidirectionality is what we call *inoutstar*. Learning rules that are more familiar from the literature are either not stochastic at all (clipped learning) or do not match the conditional probabilities in the environment (leaky learning, *instar*, *outstar*).

Learning the toy language from UF–SF pairs 4.1

Suppose we have the toy language of Section 2.1, with the coronal bias of Section 2.6: the UF |an#pa| occurs 37.5% of the time, of which the SF will be /ampa/ 70% of the time and /anpa/ 30% of the time; the UF |an#ta| occurs 37.5% of the time, yielding the SF /anta/ 100% of the time; the UF |am#pa| occurs 12.5% of the time, yielding the SF /ampa/ 100% of the time; and the UF |am#ta| occurs 12.5% of the time, yielding the SF /amta/ 95% of the time and /anta/ 5% of the time. The task for the virtual learner is to start with the network of Figure 3, but with all weights set to 0 (or a small random number), and then to adapt these weights under supervision from the language data.

For this purpose, we feed the network with a large number, say 100,000, of UF–SF pairs randomly drawn from the language environment. Thus we feed the learner with the pair |an#ta|–/anta/ in 37.5% of these 100,000 cases, and with |an#pa|–/ampa/ 26.25% of the time (70% of 37.5% is 26.25%); also with |am#pa|–/ampa/ 12.5% of the time, with |am#ta|–/amta/ 11.875% (95% of 12.5%) of the time, with |an#pa|–/anpa/ 11.25% (30% of 37.5%) of the time, and with |am#ta|–/anta/ the remaining 0.625% (5% of 12.5%) of the time. In Figure 3 we see that the five most common pairs are represented in the working network with the five strongest weights (though not in exactly the same order). The intuition, then, is that the learning algorithm should make those weights strong that connect nodes that are associated with each other in the data.

Now, what does it mean to "feed" UF–SF data to the network? It means that if at a certain point during learning we want to feed the network with, say, the pair |an#pa|–/ampa/, we set the activity of nodes 1 (|an#pa|) and 7 (/ampa/) to 1.00 and the activities of the other six nodes to 0. This is the situation in Figure 6. We then let activity settle down by having the activity spread 500 times (this does nothing in this case, because all eight nodes are clamped). After this, we change all 16 connection weights by a small amount. This whole procedure of selecting an UF–SF pair, setting the activities, vacuously spreading the activities, and changing the weights, is repeated 100,000 times, as said. In Section 4.2 through Section 4.7 we discuss six ways to do the weight changes and compare their suitability for implementing bidirectional probability matching.

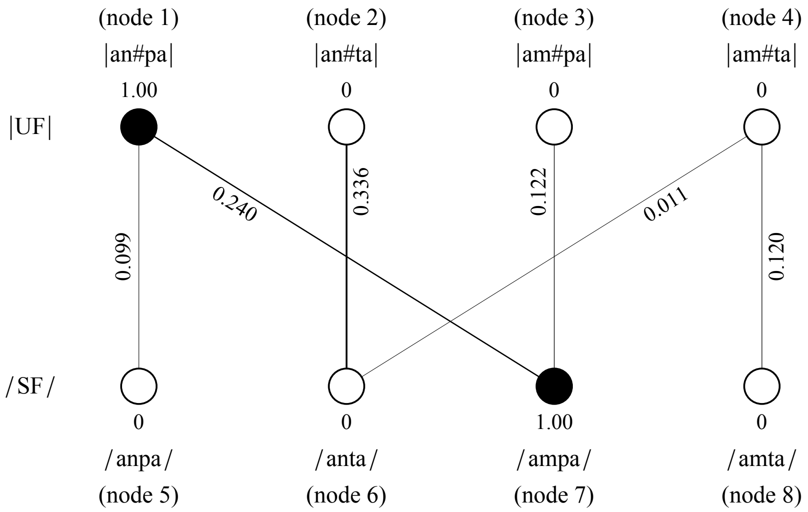


Figure 6: Supervised two-level learning: all nodes are clamped, and only one node is on in UF as well as SF

4.2 *Unbounded linear learning*

The simplest way to react to the shared activity of nodes 1 and 7 is to raise the weight of their connection ($w_{1,7}$) by a small amount, say 0.01, and not change the weight of any of the other 15 connections. This can be achieved by the following "Hebbian learning" formula:

(9) $\Delta w_{ij} = \eta_w a_i a_j \qquad (\text{for } i = 1..4, \;\; j = 5..8)$

where η_w is the *learning rate*, which is 0.01 here. This works correctly, because for $i = 1$ and $j = 7$, $a_i a_j$ equals 1 (because both a_i and a_j are 1.00), whereas for all 15 remaining i–j combinations either a_i is 0, or a_j is 0, or both a_i and a_j are 0. So $w_{1,7}$ is indeed the only weight that changes. The rule is named after Hebb (1949), who proposed that a synaptic strength increases when two neurons fire together, though he did not actually propose formula (9).

There is a problem with learning rule (9). If it goes on for 1000 times, $w_{1,7}$ will change approximately 250 to 275 times, because the network will be fed the |an#pa|–/ampa/ pair 26.25% of the time. A simulation with 2000 randomly drawn pairs is shown in Figure 7. We see that w_{ij} increases linearly with time, and goes on to do so without bounds. It has been known from the beginning of neural network modelling that the "pure Hebbian learning" of (9) exhibits this pathological behaviour (Rochester *et al.* 1956). Various devices have been

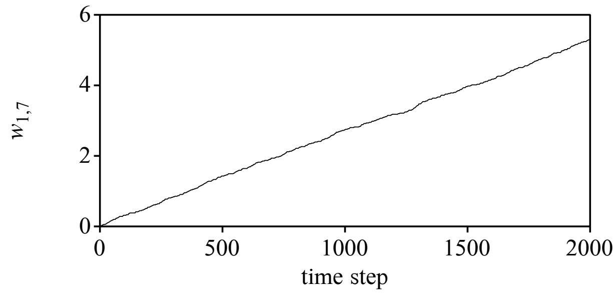


Figure 7:
The development of a weight in pure Hebbian learning: linear and without bounds

proposed in the literature to keep w_{ij} within bounds; in Sections 4.3–4.7 we discuss their suitability for our bidirectional toy case.

Clipped linear learning 4.3

A brute-force method to keep w_{ij} within bounds is to clip w_{ij} from below by a value w_{min} (e.g. 0) and from above by a value w_{max} (e.g. 1). This method has the tendency of ultimately pushing most weights toward either w_{min} or w_{max}. If the input is such that a single node i is on (and all other input nodes are off), and there are 10 output candidates (= nodes), then e.g. 3 output candidates will be maximally activated (namely those for which w_{ij} equals 1) and 7 candidates will be off (namely those for which w_{ij} equals 0). This means that under the first or third scenario from Section 2.5, three output candidates have a probability of 1/3 to win, and the remaining seven output candidates have a probability of 0 to win (the second scenario from Section 2.5 is not interpretable). This situation is not good for stochastic decision-making, where we want probabilities to move gradually from 0 to 1 or the reverse. In our simulations in Sections 5 and 6 we therefore work with activities that are not clipped from above (although they are clipped from below at 0, so we get some zero probabilities).

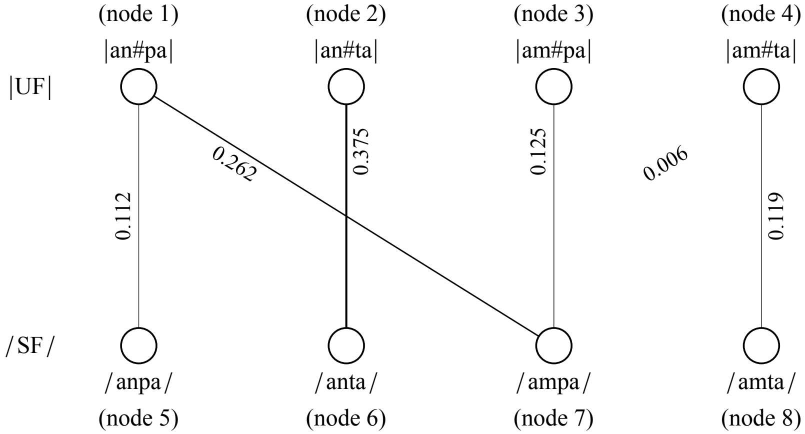
Leaky learning 4.4

A more gradual way to keep w_{ij} within bounds is to introduce leak:

(10) $\Delta w_{ij} = \eta_w \left(a_i a_j - w_{ij} \right) \qquad (\text{for } i = 1..4,\ \ j = 5..8)$

The weights now start to rise exactly as in Figure 6, but after some time they start to rise more slowly, growing exponentially toward an equilibrium in very much the same way as in Figure 5, albeit with never-ending fluctuations because of the stochasticity of the input. After many pieces of data (UF–SF pairs), the weights come to hover around those in Figure 8.

Figure 8: The average end state of leaky learning in the language environment of Section 4.1



In this final (asymptotic) situation after learning, each weight has become exactly the probability of the relevant UF–SF pair as mentioned in Section 4.1; the sum of all the weights in Figure 8 is 1. We could have predicted this result theoretically by realizing that in the equilibrium situation the expected weight change $\langle \Delta w_{ij} \rangle$ must be 0 for each connection; in other words: for each i and j the average of Δw_{ij} over all possible UF–SF pairs that could come in next, weighted by the probabilities of these pairs according to Section 4.1, must be zero. Equation (10) then tells us that the expectation value $\langle a_i a_j - w_{ij} \rangle$ will then move toward zero, so that the weight w_{ij} will ultimately go toward the correlation between a_i and a_j:

(11) $$w_{ij} \rightarrow \left\langle a_i a_j \right\rangle$$

Thus, the asymptotic behaviour of w_{ij} can be predicted if we know the statistics of the activity pattern. For instance, 26.25% of the time node 1 is on ($a_1 = 1$) and node 5 is off ($a_5 = 0$), 11.25% of the time nodes 1 and 5 are both on ($a_1 = a_5 = 1$), 62.5 percent of the time nodes 1 and 5 are both off ($a_1 = a_5 = 0$), and 0%

of the time node 1 is off ($a_1 = 0$) and node 5 is on ($a_5 = 1$); the weight of the connection between nodes 1 and 5 will therefore go to $\langle a_i a_j \rangle = 0.2625 \cdot 1 \cdot 0 + 0.1125 \cdot 1 \cdot 1 + 0.625 \cdot 0 \cdot 0 + 0 \cdot 0 \cdot 1 = 0.1125$. Since three of the four terms are zero if node 1 and node 5 are not both on, this expectation value necessarily equals the probability that both node 1 and node 5 are on simultaneously. This is a general result if all activities can take on only the values 0 and 1:

(12) $w_{ij} \rightarrow \mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)$

Such pure correlation learning looks nicely simple, but has a disadvantage. Relatively rare inputs will lead to weak connections: |am#pa| has a three times weaker connection in Figure 8 than the three times more common input |an#ta|. This disregards the perfect degree to which the SF /ampa/ can be predicted from |am#pa|. The frequency difference between |am#pa| and |an#ta| thus leads to a large difference in the activities at SF, which means that further on in processing the rare UF counts *much* less heavily than the more frequent UF. A learning rule that focuses on reliability rather than frequency alone may fare better in this respect. Another problem is that the small output activities for rare inputs (such as 0.125 for /ampa/) do not reflect the full activity that occurred during learning (which was 1 for /ampa/).

Outstar learning 4.5

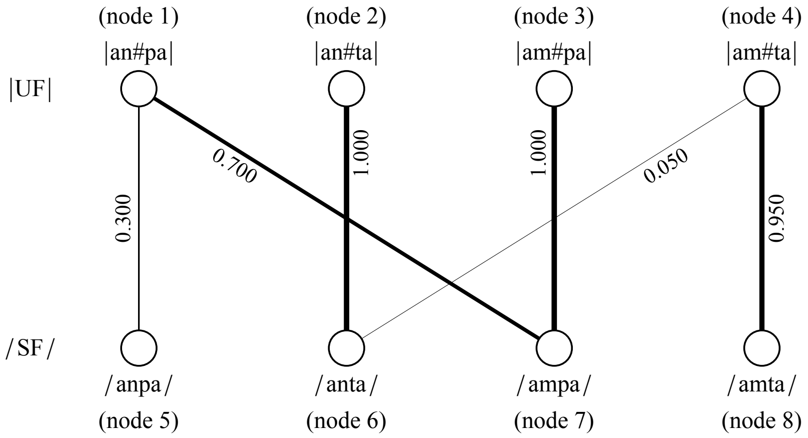
The cause of the problems with leaky learning is that the algorithm leaks *too much*: connections get weaker even if their two nodes are both inactive. One way to remedy the problem is to use the *outstar* learning rule (Grossberg 1969):

(13) $\Delta w_{ij} = \eta_w \left(a_i a_j - a_i w_{ij}\right)$ (for $i = 1..4,\quad j = 5..8$)

This learning rule does nothing with a connection if its input node is off ($a_i = 0$). A property that none of the learning algorithms discussed above share is that for outstar learning we have to assign a direction to the process, for instance to define UF as the input level and SF as the output level; so we choose the production view here, as in Section 2.

For the example in Figure 6, outstar learning will strengthen the connection between nodes 1 and 7, weaken the connections 1–5, 1–6 and 1–8, and leave the remaining 12 connections alone. After many learning steps with UF–SF pairs from our toy language, the weights come to hover around the equilibrium values in Figure 9.

Figure 9: The average end state of outstar learning in the language environment of Section 4.1



In the end, the weights turn out to have become the conditional probabilities of SF given UF (as in Figure 3), so outstar learning exhibits the probability-matching behaviour that we wanted; the sum of the weights going out from each UF node is 1. This could have been predicted theoretically, by realizing that in the equilibrium situation $0 = \langle a_i a_j - a_i w_{ij} \rangle = \langle a_i a_j \rangle - \langle a_i \rangle w_{ij}$, so if learning converges, it must move the weights asymptotically toward

$$(14) \qquad w_{ij} \rightarrow \frac{\langle a_i a_j \rangle}{\langle a_i \rangle}$$

For cases where all activities during learning can only be 0 and 1, equation (14) reduces to the conditional probability that output node j is on given that input node i is on:

$$(15) \qquad w_{ij} \rightarrow \frac{\mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_i = 1\right)} = \mathrm{P}\left(a_j = 1 \mid a_i = 1\right)$$

Outstar learning has several advantages. As the weights in outstar learning come to reflect conditional probabilities, the weights naturally stay within the limits of 0 and 1. Furthermore, outstar learning fares better than correlation learning with respect to reliability, mimicking the GLA for Stochastic OT: the connections from |am#pa| and |an#ta| are now equally strong, reflecting the fact that their SF outputs can be equally reliably predicted from the UF. Also, the activities at SF will now be 1 for these two inputs, just as during learning.

Outstar learning also has a disadvantage over the leaky learning model in (10): it loses all dependency of SF activity on the frequency of the input. A way to have both reliability and frequency influences could be to somehow combine (10) with (13). There is a problem with both (10) and (13), though: some nodes at SF, such as /anpa/, are very *specific* for certain UF forms, and this is not rewarded with a strong connection; in other words, (15) does not take into account whether or not output node *j* is on if input node *i* is off. One can look at this in terms of the reliability of the reverse process, i.e. the mapping from SF to UF in word recognition: the connection in Figure 9 from the SF /anpa/ to the UF |an#pa| is only 0.300, although the UF can be predicted with 100% reliability from the SF. We tackle this problem in Section 4.6.

Outstar learning is close to the *delta rule* of supervised learning algorithms (Widrow and Hoff 1960), where the weight update is proportionate to the *error* that the network would make when allowed to run freely (i.e. with UF clamped but SF unclamped); the error is the difference between the desired activity at SF (i.e. the number of 0 or 1, as used as a_j in the SF clamping above) and the activity that the SF node j would get when only the input UF nodes are clamped, which is $\Sigma a_i w_{ij}$ in the examples of Section 2:

(16) $$\Delta w_{ij} = \eta_w \left(a_i a_j - a_i \sum_{k=1}^{4} a_k w_{kj} \right) \qquad (\text{for } i = 1..4, \quad j = 5..8)$$

This, together with the property of probabilities conditional to the input, makes this algorithm a good candidate for replicating results previously found with Stochastic OT. This algorithm is therefore expected to be of use when in Section 6 we model auditory dispersion, a phenomenon previously modelled successfully with Stochastic OT (Boersma and Hamann 2008).

4.6 Instar learning

To take the specificity of SF (Section 4.5) into account, we can apply the instar learning rule (Grossberg 1969, 1976; Rumelhart and Zipser 1985),[3] which is the outstar learning rule in the opposite direction of processing:

(17) $\Delta w_{ij} = \eta_w \left(a_i a_j - a_j w_{ij} \right) \qquad (\text{for } i = 1..4, \;\; j = 5..8)$

This learning rule does nothing with a connection if its output node is off ($a_j = 0$). As with outstar, we explicitly have to define what the input and what the output level are (again, we take the production view, with UF as input and SF as output). For the example in Figure 6, instar learning will strengthen the connection between nodes 1 and 7, weaken the connections 2–7, 3–7 and 4–7, and leave the 12 remaining connections alone. For our toy language, the weights come to hover around the values in Figure 10.

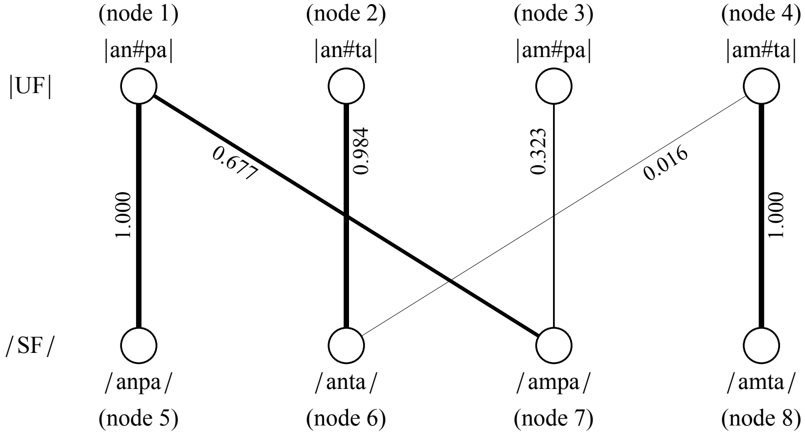


Figure 10: The average end state of instar learning in the language environment of Section 4.1

Asymptotically, the weights turn out to become the conditional probabilities of UF given SF; the sum of the weights coming in at each SF node is 1. In the theoretical equilibrium situation,

(18) $w_{ij} \to \dfrac{\langle a_i a_j \rangle}{\langle a_j \rangle}$

[3] Oja (1982) has a formulation in which the second a_j is squared.

For cases where all activities during learning can only be 0 and 1, equation (18) reduces to the conditional probability that input node i is on given that output node j is on:

$$(19)\qquad w_{ij} \to \frac{\mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_j = 1\right)} = \mathrm{P}\left(a_i = 1 \mid a_j = 1\right)$$

The two problems with rare inputs are not addressed, but the specificity problem is solved: the connection from the SF /anpa/ to its only possible UF |an#pa| has a weight of 1. The effect of the different frequencies of the different underlying forms has also returned, with the connection from /ampa/ to |an#pa| now being stronger than the connection from /ampa/ to |am#pa|, as in leaky learning but not as in outstar learning. The drawback is that the infrequent UF |am#pa| will now produce a much smaller activity pattern in SF (a total of 0.323) than the more frequent UF |an#pa| (a total of 1.677). We address this problem in Section 4.7.

Instar learning is known from work on competitive learning (Grossberg 1976, 1987; Rumelhart and Zipser 1985). This algorithm is therefore expected to be of use when in Section 5 we model phonological category creation, a phenomenon that has been partially modelled before with competitive learning (Guenther and Gjaja 1996).

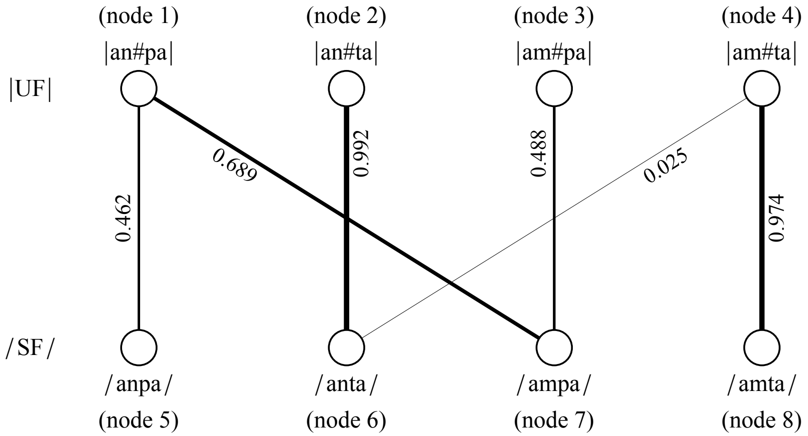
Inoutstar learning 4.7

To model category creation we seem to need unsupervised instar learning (Section 4.6), and to model auditory dispersion we seem to need supervised outstar learning (Section 4.5). However, both processes occur in the AudF–SF interface, so the same network will have to model them both. Our goal, therefore, is to model both category creation and auditory dispersion with a single learning algorithm, perhaps a compromise between instar and outstar. We call this the "inoutstar" learning rule:

$$(20)\qquad \Delta w_{ij} = \eta_w \left(a_i a_j - \frac{a_i + a_j}{2} w_{ij} \right) \qquad (\text{for } \; i = 1..4, \;\; j = 5..8)$$

This learning rule does nothing with a connection if both of its nodes are off. For the example in Figure 6, inoutstar learning will strengthen the connection between nodes 1 and 7, weaken the connections where one node is on and the other off (1–5, 1–6, 1–8, 2–7, 3–7 and 4–7), and leave the remaining nine connections alone. For our toy language, the weights come to hover around the values in Figure 11.

Figure 11: The average end state of inoutstar learning in the language environment of Section 4.1



Asymptotically, each weight turns out to become the harmonic mean of the weights of Figures 9 and 10. In the theoretical equilibrium situation,

(21) $$w_{ij} \to \frac{2\left\langle a_i a_j \right\rangle}{\left\langle a_i + a_j \right\rangle}$$

For cases where all activities during learning can only be 0 and 1, equation (21) reduces to the harmonic mean of the two conditional probabilities:

(22) $$w_{ij} \to \frac{2\ \mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_i = 1\right) + \mathrm{P}\left(a_j = 1\right)}$$

$$= \frac{2\ \mathrm{P}\left(a_i = 1 \mid a_j = 1\right)\mathrm{P}\left(a_j = 1 \mid a_i = 1\right)}{\mathrm{P}\left(a_i = 1 \mid a_j = 1\right) + \mathrm{P}\left(a_j = 1 \mid a_i = 1\right)}$$

Inoutstar learning was used before by McMurray *et al.* (2009) to simulate word–object mappings. It combines the desirable properties of instar and outstar: it tackles all problems mentioned to some extent, though none of them perfectly: it does some probability matching, it has some specificity, and it is even a bit frequency-dependent in both directions (because instar and outstar are both frequency-dependent in one direction). It has the additional advantage over both instar and outstar learning that it is symmetric in input and output: the formula stays the same if i and j are swapped, i.e. the inoutstar learning rule does not care about the direction of processing. This will even be true if there are separate weights in the beginning, i.e. if w_{ij} is not equal to w_{ji} at the beginning of learning: equation (22) shows that inoutstar learning causes the weights to become symmetric. Inoutstar can therefore be expected to implement quite well the bidirectionality of models such as the one in Figure 2.

Conclusion 4.8

A general formula for the change in the weight between input node i with activity a_i and output node j with activity a_j could be

(23) $$\Delta w_{ij} = \eta_w \left(a_i a_j - \textbf{\textit{instar}}\ a_j w_{ij} - \textbf{\textit{outstar}}\ a_i w_{ij} - \textbf{\textit{weightLeak}}\ w_{ij} \right)$$

We investigated pure Hebbian learning (*instar* = 0, *outstar* = 0, *weightLeak* = 0), leaky learning (*instar* = 0, *outstar* = 0, *weightLeak* = 1), instar learning (*instar* = 1, *outstar* = 0, *weightLeak* = 0), outstar learning (*instar* = 0, *outstar* = 1, *weightLeak* = 0), and inoutstar learning (*instar* = 0.5, *outstar* = 0.5, *weightLeak* = 0). Of these, inoutstar learning combines to some extent some of the good properties of the other learning algorithms, such as symmetry (insensitivity to the direction of processing), probability matching in both directions of processing, specificity in both directions of processing, and sensitivity to the frequency of the input in both directions. In Sections 5 and 6 we investigate the suitability of this algorithm for two hitherto separately modelled phenomena, namely category creation and auditory dispersion.

The equations from Sections 2 through 4 that we use for the simulations in Sections 5 and 6 are only the simplest ones that meet the requirements above, namely (7), (3) and (20). We summarize them here in their generalized forms that work not only for the toy example of Sections 2 through 4 but for any network with a combination of clamped and unclamped nodes, including the networks of Sections 5 and 6. As for activation spreading, every clamped node j has a constant activity a_j, and every unclamped node j starts with excitation $e_j = 0$ and activity $a_j = 0$ after which its excitation changes 100 or 500 times according to

(24) $$\Delta e_j = 0.01 \cdot \left(\sum_i w_{ij} a_i - e_j \right)$$

where the index i runs over all nodes connected to j. After each of these time steps, the activity of every unclamped node j is immediately determined from its excitation by the simple rectifying excitation function, which prevents negative activities:

(25) $$a_j = \max\left(0, e_j\right)$$

After cycling through all the time steps, the activities of all unclamped nodes should almost have settled, and the weight of the connection between any pair of nodes i and j is updated by the (symmetric and bidirectional) inoutstar learning rule:

(26) $$\Delta w_{ij} = \eta_w \left(a_i a_j - \frac{a_i + a_j}{2} w_{ij} \right)$$

5 PHONOLOGICAL CATEGORY CREATION

In this section we present a neural network that can model the emergence of simple phonological categories in the language-acquiring child. In terms of Figures 1 and 2, phonological categories, such as feature values, are present in the adult phonological Surface Form (SF). In the comprehension direction of Figure 2, the cue knowledge at the adult phonology–phonetics interface classifies the thousands of different sounds that can occur in the Auditory Form (AudF) into a small number of discrete categories at SF. In terms of neural networks, a “category” can only be defined as a stable, or “attractive”, activity pattern. That is, an adult network at the phonetics–phonology interface should “filter” the thousands of possible activity patterns at AudF into only a small number of possible activity patterns at SF.

In existing models of phonology category learning (Guenther and Gjaja 1996; Boersma *et al.* 2003) the adult state of the grammar or network comes about by training the grammar or network with a large number of auditory values at AudF, without telling the grammar or network what the intended category was. Such “unsupervised” learning is also employed here. In Section 5.3 we describe how this learning proceeds, after having described the network structure in Section 5.1 and the AudF input in Section 5.2. The resulting adult network is presented in Section 5.4 and understood in Section 5.5, after which we investigate its behaviour in perception (Section 5.6) and production (Section 5.7) and compare this behaviour to the existing literature (Section 5.8). In-depth investigations of the underlying mechanism (Section 5.9) and its response to variable environments (Section 5.10) follow. Finally, we compare the network’s performance and assumptions to the existing literature (Section 5.11).

A network for category emergence 5.1

Figure 12 shows the structure of the network that should learn the task of categorizing auditory input. The network contains only two levels of representation: the phonetic Auditory Form, which is the input for the listening learner, and the phonological Surface Form, which is the listener’s perceptual output. As we model only the phonological part of comprehension, we do not include the higher levels of Figures 1 and 2 (Underlying Form and Morphemes). Moreover, as most of Section 5 models only the comprehension direction and not the production direction (the exception being Section 5.7), we do not include the

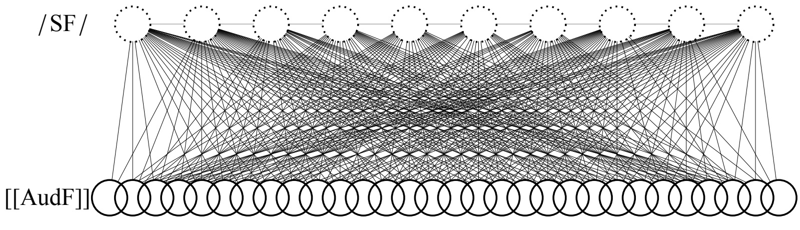


Figure 12: The initial state of a network for category creation, with continuous sound coming in at clamped AudF and discrete behaviour emerging at unclamped SF

Articulatory Form (see Section 6 for that), although including such a level would not change any of the perception or category creation simulations, as we explain in Section 5.7.

The Auditory Form represents an auditory continuum, such as the frequency spectrum along the basilar membrane. While the basilar membrane has 3,500 inner hair cells, each of which is connected to a fiber in the auditory nerve, we represent the spectrum here with only 30 nodes for reasons of visualizability (and computation time). Figure 12 arranges the nodes in a natural order, with the leftmost node (node 1) representing the lowest audible frequency of the continuum, and the rightmost node (node 30) representing the highest audible frequency.

As a simplification we allow the incoming sound to activate only one small region of AudF (as e.g. in Figure 14); this means that AudF can only represent a unitary spectral continuum, and for this we choose the spectral centre of gravity (CoG).

The Surface Form in Figure 12 will come to represent phonological "sibilant place", because that is the feature that has CoG as its main auditory correlate. Every category that the SF in Figure 12 has to be able to represent, is therefore a value of the feature sibilant place. Languages seem to have between one and four primary sibilant place values, so our SF should be able to represent between one and four categories. Even if we restrict the activity patterns at SF in such a way that each node is either "on" (1) or "off" (0), the SF in Figure 12 can represent as many as $2^{10} = 1024$ different categories; and if "on" nodes cannot be shared between categories, the SF in Figure 12 can represent 10 different categories. In either case, our 10 nodes should be more than enough to represent any number of feature values between one and four in a distributed way.

As can be seen in Figure 12, AudF and SF are fully connected to each other: there are 300 connections between them, one for each pair of AudF node and SF node. Initially, these weights are small and random: uniformly distributed between 0 and 0.1, as shown as black lines in the figure. This randomness is meant to ensure that in its initial state the network is poor at classifying incoming sounds into stable categories: in perception (with a clamped AudF and an unclamped SF, as in Figure 12), any local activity peak in AudF will just lead to a small and random pattern at SF (as can be seen for example in Figure 14).

As illustrated in Section 5.4, this situation will change when the network learns from incoming sounds at AudF: the weights will become larger and less random. As Section 5.6 shows, the result is the desired emergence of categorical behaviour in the network.

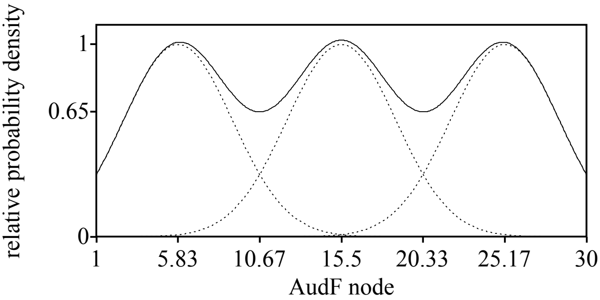
Finally, Figure 12 shows 45 connections within SF: one for each pair of SF nodes (i.e. not just between nodes that happen to look “adjacent” in the visually one-dimensional set-up of Figure 12). These connections have negative weights of -0.1 (shown in light gray) in order to make sure that the SF nodes inhibit each other’s activities. As a result, learning causes the SF nodes to become connected to different AudF patterns, which is illustrated in Section 5.4 and explained in 5.5. This ensures that different categories from the network’s language environment lead to different categorical patterns in the learner’s own SF. This mutual inhibition is a mechanism we borrow from competitive learning models (Grossberg 1976, 1987; Rumelhart and Zipser 1985). The negative weights do not change during learning.

An input distribution for sibilant place 5.2

As said, the network will be trained with the auditory distribution alone, i.e. it will have to learn from incoming CoG values from a language environment, without supervision. Thus, the virtual learner repeatedly hears an incoming sound but is never told to what category it belongs and is never told any of the associated higher levels of representation, such as meaning. Neither is the learner told how many categories the language has.

For the coming sections of this paper, we investigate a very simple language environment that consists of three sibilant fricatives, namely /ʂ/, /ɕ/ and /s/, as in Polish or Mandarin. The spectral centre of gravity of each sibilant is distributed according to a Gaussian distribution, as in the three dotted curves in Figure 13. The distance between the peaks is one third of the range of the continuum, i.e. 9.667 nodes, and the standard deviation of each peak is one third of that (i.e. 3.222 nodes). The three sibilants are equally frequent in the language environment, so that the total distribution of CoG values is the solid curve in Figure 13.

Figure 13:
A CoG distribution in a language with three sibilant places



The beginning learner does not yet know that there are three curves; she only hears input tokens one by one without category labels, and the summed distribution of these input tokens gradually and incrementally grows toward the total CoG distribution. The valleys in this curve are rather shallow, namely approximately 64% of the average height of the three peaks. In the end, it is on the basis of input drawn from the summed distribution, with its shallow valleys, that the learner will have to figure out that there are three categories.

5.3 *Unsupervised learning from the distribution*

A full description of a language learning procedure involves describing how each input is applied to the learner, how the learner processes this input, and how the learner then changes her grammar. In our case, the input to the network is formed by the learner's language environment repeatedly producing a single CoG value randomly drawn from the summed distribution (equivalently, the language environment randomly selects one of the three sibilants, then randomly draws a CoG value from that sibilant's Gaussian distribution; the important restriction is that the learner is not told which sibilant was selected). The learner receives this CoG value as an activity at AudF, then processes it by spreading this activity to SF, and finally updates the connection weights between AudF and SF on the basis of the activities at AudF and SF. We will show here that after 20,000 or so incoming CoG values, this procedure leads to the emergence of categorical behaviour at SF.

Whenever a CoG value is applied to AudF, this produces an activity pattern at AudF of the form shown in Figure 14. The CoG value

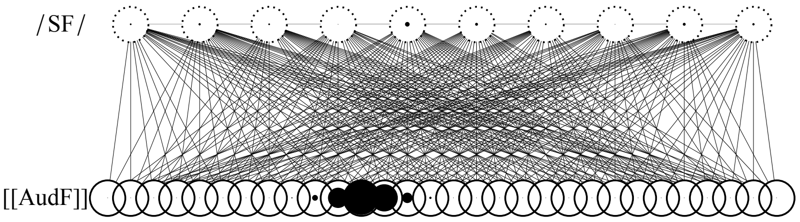


Figure 14: Applying an input

is an (unrounded) node number between 1.0 and 30.0. In Figure 14, the CoG value is 12.3. The nodes in the vicinity of location 12.3 are then activated according to a Gaussian shape with a height of 1 and a standard deviation of 4 percent of the extent of the continuum (i.e. $0.04 \cdot 29 = 1.16$ nodes), mirroring the width of a region of activity on the basilar membrane.[4] This activates node 12 most strongly (at a distance of 0.3), node 13 a bit less strongly (distance 0.7), node 11 (distance 1.3) even less strongly, and so on; the activities of nodes further away than nodes 14 and 10 are too weak to be visible in the figure. Independently of whether the centre of the Gaussian bump is located on a node or somewhere between two nodes, the total activity in AudF is always around 2.908 (if the CoG value is very close to the left or right edge, the total activity is less, because a part of the bump is cut off).

After the input is applied to AudF, the AudF nodes in Figure 14 are clamped (as shown by the solid edges of their circles), i.e. their activities are kept at the applied values (those seen in the figure) throughout the spreading of activities. The SF nodes, by contrast, are unclamped (as shown by their dotted circumferences), i.e. their activities adapt to the activities of the AudF nodes as well as to the activities of other SF nodes throughout the spreading of activities. The activities at SF start at zero, after which the activities of AudF excite the nodes at SF according to equation (24) (with positive w_{ij}); as SF activity grows, the SF nodes start to inhibit each other, again according to equation (24) (with negative w_{ij}). These excitations and inhibitions occur with a spreading rate of 0.01, with the summation in (24) running over all AudF and SF nodes. The computation of activity from excitation fol-

[4] If node 1 is at 16 ERB (1095 Hz), and node 30 is at 33 ERB (9611 Hz), then this standard deviation is $0.04 \cdot 17 = 0.68$ ERB.

lows equation (25): the activities are clipped from below at zero (i.e. negative activities are not allowed, but large positive activities are). Spreading goes on in this way for 100 time steps. The result is that ultimately the whole network would move toward equilibrium, if the spreading were not truncated after 100 time steps.

After activity spreading, the network is allowed to learn by the inoutstar rule, i.e. equation (26) applied to all 300 connections between AudF and SF, with a learning rate of $\eta_w = 0.01$. There is only one learning step per incoming CoG value.

5.4 *Result after learning: the perception of three categories has emerged*

After 20,000 incoming CoG values, the weights of the network have become those in Figure 15. At SF, nodes 2, 6 and 9 (i.e. the three that are on in the figure) have become associated to low ([ʂ]-like) CoG values, nodes 4, 5 and 8 to intermediate ([ɕ]-like) CoG values, and nodes 1, 3, 7 and 10 to high ([s]-like) CoG values. In other words, each node at SF has specialized in one of three areas of AudF, and each of these three areas of AudF is associated with approximately one third (i.e. three or four) of the SF nodes.

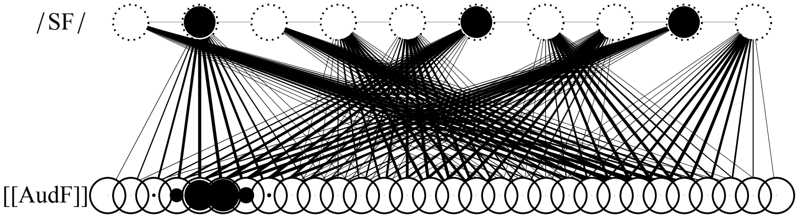


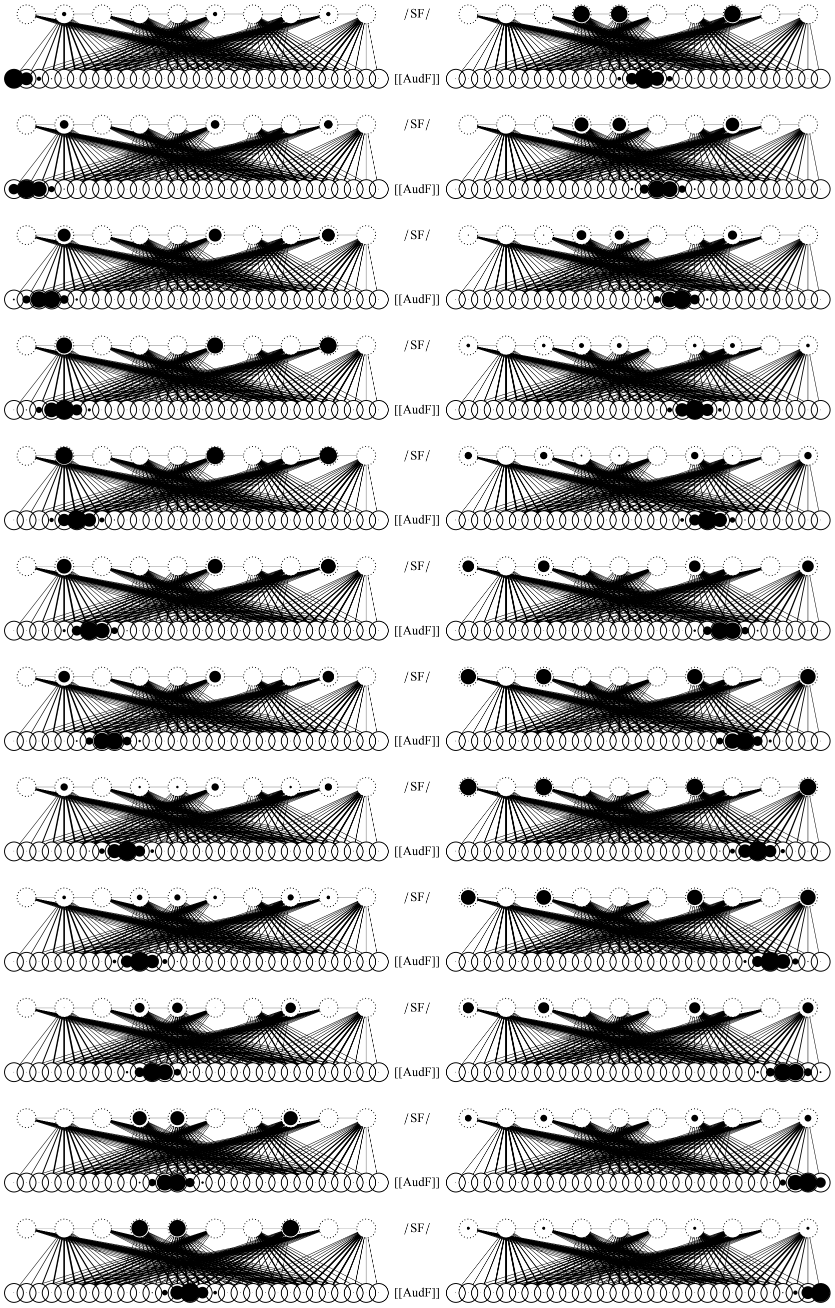
Figure 15: A network that has been trained on three peaks and has thereby become capable of categorizing

This situation of dedication of SF nodes to AudF areas causes the trained network to *behave categorically* in perception. We can see this by applying a large number of different input patterns to AudF and examining the resulting output patterns at SF. In Figure 16 we pace a local activity pattern through the whole auditory continuum from the lowest values (top-left picture) to the highest values (bottom-right picture). We see that the output at SF favours exactly three patterns of activity. For any low auditory value, only SF nodes 2, 6 and 9 switch on; for any mid value, only nodes 4, 5 and 8 switch on, and for any high value, only nodes 1, 3, 7 and 10 switch on. Since activity patterns are the brain's way of representing behaviour, the favoured 2–6–9, 4–5–8 and 1–3–7–10 patterns at SF represent favoured (or "attractive", or "stable") types of behaviour at SF, or, in other words, three *categories* (when the information proceeds up toward Underlying Form, Morphemes, and perhaps higher semantic areas of the brain, there will still be only three types of behaviour in those higher regions, because according to the adjacency property illustrated in Figure 1, those higher levels of representation cannot "look through" SF toward AudF). We can therefore call the first favoured behaviour at SF the "2–6–9 category"; it replicates the /ʂ/ category of the language of the parents. Likewise, the 4–5–8 category represents the parents' /ɕ/ and the 1–3–7–10 category represents the parents' /s/.

The final network of Figure 15 differs from the networks we discussed in Sections 2 through 4 in that the network of Figure 15 no longer represents a phonological category as a single node, but represents phonological categories in a *distributed* manner, namely as two or three SF nodes each. The same is true of AudF: every incoming sound activates more than one node at AudF. A biologically desirable property that such a network displays is *redundancy* in the representation of patterns: if a couple of AudF nodes die, and one SF node dies, the network will still perform its classification task quite well. In Figure 15, for example, every incoming sound will still generate one of three stable patterns at SF. For purposes of category creation, it is even more important that having 10 SF nodes allows any number of categories to be created: rather than forcing the existence of 10 categories, as would be the case for the networks in Sections 2 through 4, the 10 nodes are divided roughly equally among the two or three or five categories that the peaky language distribution suggests there are.

We conclude that there come to be three types of stable behaviour at SF, to be interpreted as three phonological categories. This categoricality comes about gradually during learning. On the way to the final state of the network, the categoricality of the behaviour increases from nothing (the random behaviour at SF that the network of Figure 14 exhibits) to almost perfect (the behaviour of the eighth picture in Figure 16, which has the same input). Thus, **categoryhood is gradient** in this model: during development, the patterns gradually grow from

Figure 16: Pacing the trained network through the Auditory Form yields three types of patterns in the Surface Form



being less attractive to being more attractive, without there being a moment at which one can say that a category has just come into existence. During the acquisition period, the behaviour therefore changes from random via slightly categorical toward very categorical.

How does category creation work? 5.5

After seeing *that* category creation works, we would like to understand *why* it works.

The most crucial aspect of the network is the competition at SF. This is known from competitive learning models (Grossberg 1976, 1987; Rumelhart and Zipser 1985; Guenther and Gjaja 1996), which typically implement competition by "manually" setting the most active output node (the "winner") to an activity of 1 and all other nodes (the losers) to an activity of 0. This winner-takes-all procedure is an extreme version of what we use in this paper, and could be implemented in our case as follows: if after 100 steps of activity spreading to SF (as in Figure 14) we drastically severed all connections between the SF level and the AudF level, and thereby allowed activity to spread only between the nodes of SF, then the inhibitory connections within SF would reduce the activities of all nodes as long as more than one node were on; one by one, the weakest nodes would drop to zero activity, and this reduction would stop when only a single node were left, which would have some nonzero activity remaining; this node would be the one that had the highest activity to start with. Our exhaustive inhibitory connection scheme, which does not use winner-takes-all, can be seen as a gradual version of the original competitive learning models; it is a more "automatic" version of competition, because no artificial temporary connection severing is necessary; still, the competition is guaranteed by the existence of inhibitory connections within SF.

In the original competitive-learning models, the winner-takes-all step is followed by a learning step in which the weight(s) of the connection(s) between the active input node(s) and the winner are increased and the weights of the connections between the inactive input nodes and the winner are decreased, a procedure identical or similar to instar learning. Our gradual version of competitive learning with inoutstar learning creates distributed categories by the same cause, which we try to explain now.

First imagine that there is only one node at SF. In Figure 14 this node will be active whenever a part of AudF is switched on. The connections from this node to AudF regions that are often on will strengthen more than the connections to AudF regions that are rarely on. After some time, the connection weights for the various AudF nodes will come to follow a pattern similar to the summed curve in Figure 13. This means that if we pace through AudF as in Figure 16, the activity of the single SF node will go up and down along with the peaks in the summed distribution. Hence, activity in the single SF node will be highest at the three tops of Figure 13. Imagine now that there are 10 nodes at SF, but there is no inhibition between them. Every node at SF will come to be connected to AudF in the same way as the single SF node in the previous imaginary network. Consequently, each node will be activated by AudF according to the summed curve in Figure 13. Imagine finally that an inhibition between all the nodes at SF is introduced. This inhibition militates against different SF nodes being on at the same time. As a result, assuming small random differences in activities between SF nodes (caused by the different random initial weights), different SF nodes will come to specialize in different regions of AudF, so that they can be on at different times (the sum of all activities at SF will still follow Figure 13; see Figure 17). A further question is: why does an SF node specialize in a contiguous *region* of AudF, rather than, say, in the left half of the first peak and the right half of the second peak? This is because of the width of the activity on AudF: the left half of the first peak tends to be active when the

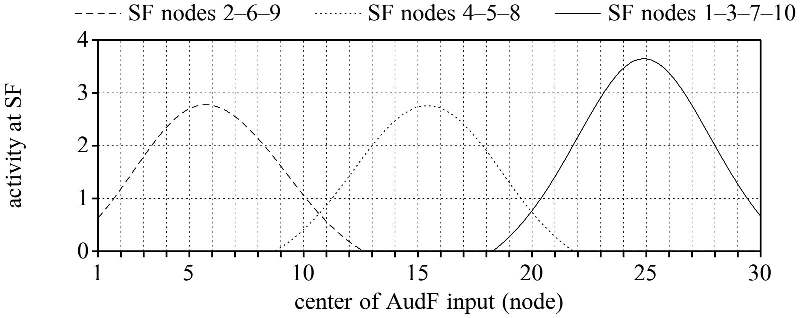


Figure 17: The degree of activation of each of the three categories, as a function of the auditory input

right half of the first peak is somewhat active as well. In other words, (spectrally) adjacent nodes at AudF have correlated activities, just as (spatially) adjacent hair cells on the basilar membrane do. If in our simulations we had instead activated only the node nearest to the selected CoG, no categorization of regions would have occurred.

The assignment of each SF node to an AudF region is not random: in fact, the SF nodes tend to become equally divided between the three categories. If each SF node were independently tuned to a region of its choice, we would find that in 5.2% of the experiments an ambient category would be presented by 0 nodes. We never find this; the division 4–3–3 is by far the most common. The cause of this equal division is the inhibition.

Investigating the network's detailed perceptual behaviour 5.6

In Figure 16 we can see that when the incoming sound paces through the auditory continuum, the degree of the activities within a category at SF is not always the same. The activities of the 2–6–9 (/ʂ/) category are much higher if AudF node 6 is on (where the peak of the first category is located, as can be seen in Figure 13) than if AudF nodes 2 or 10 (where the margins of the first peak are located) are on. Thus, the first category is much more strongly activated by the relatively common AudF patterns around node 6 than for the less frequent AudF patterns around nodes 2 and 10.

At the category boundaries, a mixed type of behaviour appears. For AudF nodes around 10 and 11, SF shows a combination of the 2–6–9 (/ʂ/) category and the 4–5–8 (/ɕ/) category: apparently, both categories are activated to some (small) extent. Observationally, this situation can correspond to an uncertainty in the listener about what the category is; an interpretation of this is that the SF candidates /ʂ/ and /ɕ/ both move on toward UF, activating in the lexicon words with underlying |ʂ| as well as words with underlying |ɕ|. Since AudF node 11 can indeed represent either of two categories from the language environment (speakers produce such auditory values sometimes when intending /ʂ/, sometimes when intending /ɕ/), such uncertainty is adaptive and appropriate (e.g. the Ganong effect mentioned in Section 1.3.2). Something similar happens for AudF nodes around 20 and

21: the listener's reaction at SF is a mixture of the 4–5–8 (/ɕ/) and 1–3–7–10 (/s/) categories.

Figure 17 shows how strongly every possible location of the Gaussian input bump at AudF activates each of the three categories at SF (after 100 spreading steps, with a spreading rate of 0.01). Thus, a bump centred at AudF node 10 causes activities of approximately 0.37 in nodes 2, 6, and 9, so that the summed activity for category 1 (= nodes 2–6–9) is 1.1, as shown in the figure. Likewise, category 2 (= nodes 4–5–8) has a summed activity of 0.4 in its three nodes, and category 3 has no activity for AudF node 10 in any of its SF nodes 1–3–7–10. In Figure 17 the activity was measured for 581 centre locations, namely for AudF nodes 1 to 30 in steps of 0.05 node.[5] The peak is higher for category 3 than for the other two categories, because this category is formed by four SF nodes instead of three.

The activity curves follow the input distributions of Figure 13 closely, with the tops at approximately the same locations. A difference with the distributions is that the activities go to zero at a distance of approximately 7 nodes from the tops. This is due to the inhibitory behaviour of the negative connection weights within SF, which e.g. renders the excitation of category 1 negative for all AudF locations above 13. The zero values then follow from the clipping mentioned in Section 5.3.

If we interpret the activities of Figure 17 as relative probabilities of perceiving a certain incoming AudF as any of the three categories (Section 2.5), we can draw the *identification curves* of Figure 18. These curves tell us how likely any incoming AudF is perceived as category 1, 2 or 3. For each category, the curve is computed by dividing the activity curve for that category (Figure 17) by the sum of the three activity curves.

The shapes of the identification curves are similar to those found with human participants in the lab; for this reason, Figure 18 labels the three categories with the language-specific phoneme labels that human participants would have to choose from (a difference with the

[5] The smoothness of the curve shows that there is no major influence of the discretization of the input continuum on the activity curves. This desirable behaviour is caused by the fact that the bumps at AudF have a Gaussian shape. With different input shapes, the activity curves at SF may display ripple.

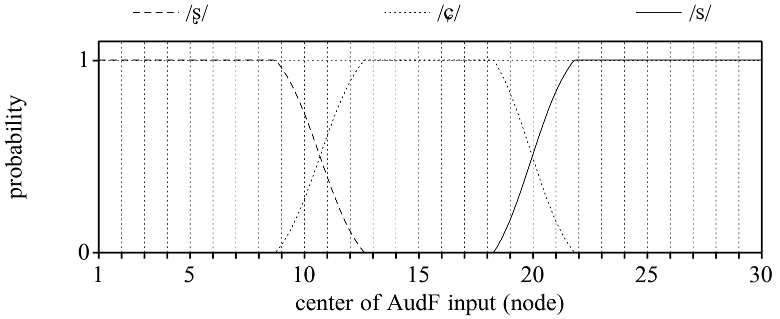


Figure 18: Identification curves after distributional category learning

human curves is that the curves in Figure 18 go to their extreme values abruptly; this difference vanishes when we realize that sounds played in the lab are supplied with transmission noise before they are converted to AudF values in the listener; another difference is that the extremes in Figure 18 are exactly 0 and 1, which is because we assumed a perfect reporting mechanism).

In the lab, humans can report not only the category they think they hear, but also how good the sound heard is as a token of that category. Such *goodness judgments* can be thought of as following the curves in Figure 17: if the listener has access to an inspection device that computes the total activity of a category at SF,[6] she will be able to calculate any activity value in Figure 17 and trivially employ that value as a reportable category goodness between 0 (poor fit to the category) and 1 (perfect fit). Relatedly, since the peaks of the curves in Figure 17 are at or near the most frequent exemplars of the categories (Figure 13), the best exemplars in a *prototype task* will be those same most frequent exemplars (this statement will be amended in Section 6.5).

Investigating the network's behaviour: production 5.7

The network is bidirectional, so it can be used to model not only perception, as in the previous section, but production as well. To measure

[6] A goodness computation for e.g. the 2–6–9 category of Section 5.4 can be performed by a simple network connected to SF, with connection weights of 1 to SF nodes 2, 6 and 9, and connection weights of 0 to the other seven SF nodes. Follow-up simulations by Chládková (2014) have shown that such weights are learnable in a three-level BiPhon model.

the production of a category, we can clamp the SF nodes of that category (i.e. nodes 2–6–9 or 4–5–8 or 1–3–7–10) at an activity of 0.8 and compute what the activity at AudF will be after 100 spreading steps. The three results are in Figure 19.

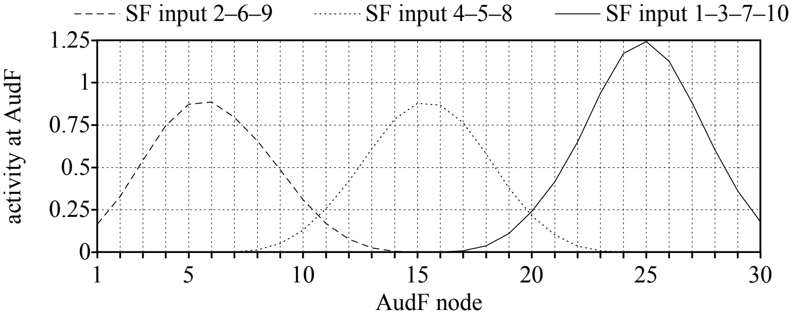


Figure 19: The activity at AudF, as a function of a three- or four-node input at SF

The learner turns out to produce the categories in much the same way as her parents, if the activities of Figure 19 are interpreted as relative probabilities. As a result of the inhibition, the standard deviation is somewhat smaller than that of the parents, but this will be counteracted (as it was in the OT model by Boersma and Hamann 2008) by the transmission noise that has to be added to the AudF values drawn from Figure 19 once we want to model multiple generations of learners.

The result in Figure 19 is not realistic. Considerations of articulatory effort will shy the learner's production away from the edges. We can model this with the network in Figure 20, in which the influence of the sensorimotor knowledge and the knowledge of articulatory effort is summarized (and extremely simplified) as a single clamped ArtF node that has strong inhibitory connections to peripheral AudF nodes and weak inhibitory connections to central AudF nodes. If the inhibitions follow a parabola, with a weight of –0.1 in the centre and –1.6 at the edges, the AudF output of the 2–6–9 category will be that shown in Figure 20.

The AudF activity for all three categories is summarized in Figure 21. The auditory realizations of the two outer categories now avoid the edges: when compared with Figure 19, their peaks slightly moved inward, and their medial tails are much longer than their lateral tails.

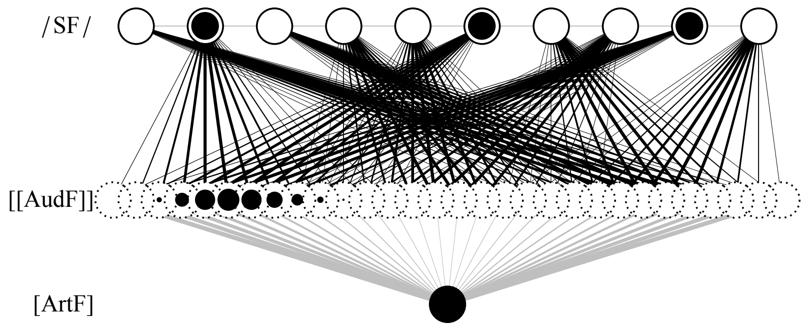


Figure 20: Network for production

This means that the learner will on average produce rather more central AudF values than her parents.

If the sound shift of Figure 21 goes on for a number of generations, the three peaks will come so closely together that a new learner cannot create three categories any longer. Inevitably, iterated learning with the procedure of Section 5 must lead to merger. However, information from above SF will come to the rescue, as Section 6 will show.

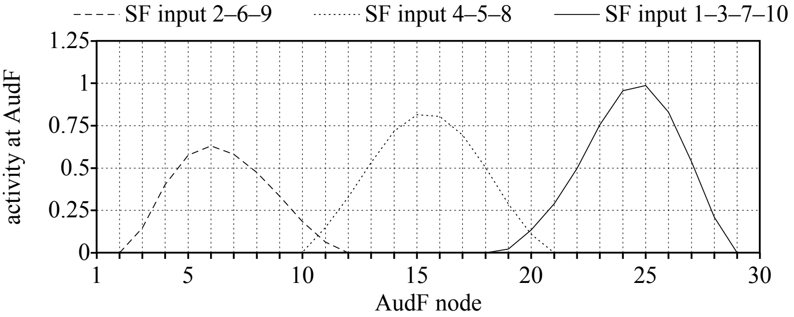


Figure 21: Production influenced by articulatory effort

It is important to note that the network of Figure 20 is compatible with the results of the two-layer network of Figure 12. That is, the network of Figure 20 works in exactly the same way as that of Figure 12 for the purposes of Sections 5.4–5.6 (and also Sections 5.8–5.10), because adding an articulatory representation below AudF cannot influence the perception process in our simulations, where the auditory representations, which lie in between the higher and the articulatory representations, are clamped (held constant) during activation spreading. An interesting variant of our simulations would appear if we let the auditory representations settle freely instead (as in e.g. McClelland and Elman 1986), in which case their connection to the articulatory representations (i.e. sensorimotor knowledge) will slowly (during activation spreading) move the auditory representations toward gestures that the listener finds easy to pronounce, which again will influence the higher (e.g. phonological) representations. With this interactive scenario, low-level perception from AudF to SF would partly go *through* articulatory representations, without articulatory representations having to lie *between* AudF and SF. Hence, several phenomena that have been brought forward by proponents of motor theory (Liberman and Mattingly 1985) or direct realism (Fowler 1986; Best 1995) in favour of articulatory representations mediating between AudF and SF can also be explained when articulatory representations lie outside the direct AudF–SF path, as was pointed out by Boersma (2012). Simulations of such phenomena fall outside the scope of the present paper. The main point we want to make here is that the network of Figure 20, not that of Figure 12, is the complete network that exhibits all the properties discussed in Sections 5 and 6.

5.8 *Replicating experimental data: categorical perception*

It is known that listeners can more easily discriminate two auditory forms that map to different phonological categories than two auditory forms that map to the same category (Liberman *et al.* 1957). The trained network of Figure 15 can replicate this behaviour, under the assumption that a listener's report whether two sounds are the same or different rests on her inspecting her SF, not her AudF. That is, when responding to the task of reporting whether two sounds are the same or not, the listener is actually reporting how different she judges the two surface forms instead.

To replicate this with the network of Figure 15, we first compute the average absolute difference between the activities of the SF nodes in the first two pictures in Figure 16. Node 1 (at SF) is activated equally (namely, 0) in both pictures, but node 2 is activated a bit more (by 0.2) in picture 2 than in picture 1. On average, the activity of a node in picture 2 differs from the activity in a node in picture 1 by an amount of 0.03. The difference between picture 3 and picture 4 is even smaller, namely less than 0.01. The difference between picture 6 and picture 7 is much larger, namely 0.05, because many nodes switch on or almost off when going from picture 6 to picture 7. Figure 22 displays all the 19 differences. It can be seen that the difference between the SF activities for adjacent AudF nodes around the category boundaries is much greater than the difference between the SF activities for adjacent AudF nodes around the category centres. This *discrimination curve* illustrates the categorical perception effect as originally observed by Liberman *et al.* (1957).

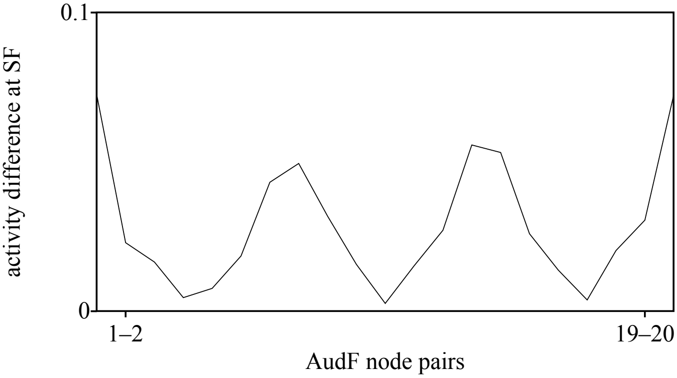


Figure 22:
The discrimination curve.
The peaks at the edges represent the difference between nodes 1 and 20

Why and when does this work? 5.9

Now that the mechanism is more or less understood, we like to know the circumstances under which the category creation procedure succeeds or fails. That is, for what kind of input data (valley depth, number of categories) do our results hold? How sensitive are our results to the hyperparameters of the network, such as the number of nodes and the amount of inhibition? Are the elements of our network design, such as rectification and the inoutstar learning rule, crucial to our results?

Valley depth 5.9.1

The first question is about the data themselves. Not all one-dimensional auditory continua come with the pooled distribution of Figure 13, i.e. with a valley depth of 0.64. If we reduce the standard deviation of the ambient peaks in Figure 13 from one third to one quarter of the distance between adjacent peaks, the depth of the valleys becomes 0.27, and the network learns as well as before (and on average slightly faster), coming up with three clear categories (4–3–3 or 4–4–2) in all (20 out of 20) replications; the same goes for data with a valley depth of 0.02. This is not surprising: sharper peaks yield better category discriminability, so we expect better learning, if anything. On the other hand, raising the standard deviation of the data to 40% of the peak distance increases the valley depth to over 0.80, and our network no longer learns equally fast: in half of the replications, the situation after 20,000 data is four insecure categories that continuously slide into each other while scanning; however, this is simply a common intermediate learning stage (also often seen after 10,000 data in the simulations of Section 5.4), and correct triple categorization always emerges when we continue to train the network toward 100,000 data (two of the four categories gradually merge). Generalizing, we can say that our network can learn three categories if the distribution shows any visible valley, although learning is faster if the valley is deeper.

To see whether having a valley-depth cut-off is bad, we compare our network to results from the literature with human subjects. Experiments that showed distributional learning have usually been performed with only two peaks, with a valley depth of 0.25 (e.g. Maye *et al.* 2008). For two categories with the same standard deviation as in Figure 13 (i.e. with a much deeper valley of 0.16, because the peaks are spaced 14.5 nodes apart), our network always (in 20 out of 20 replications) succeeds in learning the two categories perfectly, with 5–5, 6–4 or 7–3 divisions of SF. For two categories with a valley depth of 0.65 (i.e. a standard deviation 1.5 times that of the peaks in Figure 13), our network also learns well, although in a minority of replications it does so via temporarily (after 20,000 data) having a small additional category on the shoulder of a big category (when learning continues toward 100,000 data, this shoulder category merges with the main one, so this is just a sign of the expected slower learning). For valley depths used in experiments with human participants, our network therefore performs well.

5.9.2 Number of categories

In Section 5.1 we asserted that our network should be able to learn languages that have between one and four categories along the continuum. In Section 5.4 we saw that our network learns three categories from a three-peaked distribution, and in Section 5.9.1 we saw that it learns two categories from a two-peaked distribution. When confronted with a single broad peak, the network becomes a partly "auditory" listener, with continuously changing output patterns when we scan along the continuum, and with a discrimination curve, rather different from that of Figure 22, that has either a peak in the middle or two peaks around the middle (closer together than in Figure 22). This variation between learners might mirror to some extent the behaviour of participants confronted with monomodal distributions in a distributional experiment, although we can make no numerical comparisons at this point.

Our network works well for four categories with the same valley depth of 0.65 as in Figure 13, i.e. with a standard deviation of 3/4 of that of the three peaks in Figure 13: the learner divides up SF as 3–3–2–2 or sometimes 4–2–2–2, and has three discrimination peaks. With five categories, most of the learners show so much overlap between some adjacent categories that their discrimination curve has only one or two peaks instead of four; this loss of categorizability is OK, because we know of no languages with more than four CoG categories, which is why we designed our network with only 10 nodes at SF (Section 5.1).

Number of nodes at SF 5.9.3

With 10 nodes at SF, our network can learn up to four categories reliably (Section 5.9.2). It is interesting to see whether the limit of four is inherent to our type of network, or whether more categories can be learned if we modify its hyperparameters. And indeed, when we raise the number of SF nodes to 30, we can stably create up to seven categories (four categories do e.g. 7–7–6–4, with 6 nodes unconnected; seven categories do e.g. 5–4–4–4–4–4–3, with six clear discrimination peaks), under the condition that the valley depth is kept low enough to compensate for the increased effect of the 1.16-node smearing on the basilar membrane (Section 5.3). Learning eight categories usually works perfectly, but slightly fails in a minority of cases (overlapping patterns at SF for adjacent categories; e.g. one of the nodes stays on for two categories, so that one of the seven discrimination peaks is lower than the others), apparently because even with very low ambient standard deviations, i.e. valley depths close to 0, the pooled distribution of activities at AudF comes to show rather shallow "basilar valley depths". With a much greater granularity both at AudF and at SF, namely with 100 AudF nodes and 100 SF nodes, up to ten categories can be learned. We conclude that our number of SF nodes (namely, 10) limits the number of categories to 4, and the physical characteristics of our auditory continuum (namely, 0.68 ERB of basilar spreading; see footnote 4) limits the number of categories to 8 or 10.

These numbers do not seem to contradict any known fact about phonological inventories. For instance, a language with four vowel heights will have their F1 values spaced 2.1 ERB apart, if high vowels have an F1 of e.g. 300 Hz (7.3 ERB) and low vowels have an F1 of 800 Hz (13.6 ERB), and the mid vowels are equally spaced between them, i.e. at 9.4 and 11.5 ERB. This 2.1 ERB is approximately how far the peaks are spaced in our simulations with eight categories (namely, a 17-ERB range divided in eight equal steps). This can explain why languages with five vowel heights are very rare. Precise numerical fits with vowel data will have to be relegated to future work.

5.9.4 Inhibition within SF

The simulations of Section 5.4 use an inhibitory weight of 0.1 between nodes at SF. This value of 0.1 works for a great variation of valley depths and numbers of categories in the data, and for quite varying numbers of SF nodes. The value itself is not robust. If we lower this inhibitory weight to 0.01, then all SF nodes are excited equally (no off-and-on pattern as in the simulations above), and this same egalitarian pattern appears in exactly the same way for any AudF, so the number of categories created is zero (or, equivalently, one). If we raise the inhibitory weight to 1 or 2 or 10, then in the great majority of replicated simulations four categories emerge, and each of those categories has only one node at SF; that is, six of the ten nodes will never switch on. An inhibitory weight of 0.05 will generally work well, but may cause, in a sizable minority of learners, a bit of overlap in SF patterns of adjacent categories, and differences in the heights of the two discrimination peaks (in Figure 22 the two peaks are equally high). An inhibitory weight of 0.2 makes the two-category case poorly learnable (e.g. three discrimination peaks). Thus, the inhibitory weights have to have a value around 0.1, not very well allowing a factor of 2 off in either direction. We speculate that this fine-tuning of excitability of neurons may well correspond to something in biological neural systems, which function best in a state somewhere between anaesthesia and epilepsy.

We can conclude that distributional learning on a single continuum is a bit brittle in our network (in the sense of requiring a notable valley in the distribution of CoG values, and some tuning of the degree of inhibition), an observation that corresponds to what is found in a literature overview on experiments with human subjects (Wanrooij 2015: 35). In real-life acquisition, there will usually be multiple auditory continua, plus contextual information, which could make category learning easier even in cases where distributional valleys are shallow or even non-existent.

The rectifier 5.9.5

The activation function follows (3) or (25), i.e. the activity of a node is always made non-negative. An arguably simpler activation function is the identity function, as in (2), i.e. the activity of a node equals its excitation. It turns out that simulations with such an identity activation do not display stable category creation. To understand this, consider a network with identity activation that is in an initial situation in which the input to SF node 1 is 0.1 (e.g. the weights times activities of all the AudF nodes to SF node 1 sum up to 0.1, which is a possible result of initial weights being uniformly distributed between 0 and 0.1), and the inputs to all other nodes are 0 (which can happen, for instance, if all weights from AudF to these other SF nodes are zero). After the first activation spreading step with a spreading rate of 0.01, the activity of SF node 1 will be 0.001, and the activities of all other SF nodes will still be 0. After the second step, the activity of SF node 1, according to (24), will be 0.00199, but all other SF nodes will be inhibited by SF node 1, i.e. each of their activities will be $-0.1 \cdot 0.01 \cdot 0.001$ (the inhibitory weight times the spreading rate times the activity of SF node 1 after one step) $= -0.000001$. After infinitely many steps, the activity of SF node 1 will be 2/19, and that of each other SF node will be −1/171.[7] This example serves to illustrate that if we allow negative activities, such negative activities will actually occur. Combined with inhibitory weights, the occurrence of negative activity results in increasing the positive activity of other nodes, defeating the whole idea behind inhibition. This has happened here to SF node 1, which with the rectifying activation function would have ended up with an activity of only 0.1 instead of 2/19. Also, negative activities defeat the purposes of inoutstar learning, including the idea that weights reflect something close to a conditional probability; with negative activities, weights can easily fall below 0 or rise above 1, and in our simulations with identity activation they indeed tend to do so without bounds, leading to chaotic restructurings of the network upon each learning step. We conclude that no stable learning is possible with an identity activation function, while stable learning is possible with the next simplest activation function, namely the rectifying activation function used throughout Section 5 and 6.

5.9.6 Learning rules

We have seen that category creation works well with the inoutstar rule. It does not work with the simpler outstar learning rule: weights and activities blow up. However, category creation turns out to work well with the equally simpler instar learning rule, if we assume that AudF is the input and SF is the output. This could be expected on the basis of earlier competitive learning studies. The only reason why we use inoutstar instead of instar is that it is symmetric and can therefore handle the cases of Section 6 as well as those of Section 5.

5.10 *Plasticity*

After learning three categories in her native language environment, the learner might move to an area of the world where four categories are spoken. The network turns out to adapt itself accordingly. If the middle category has four SF nodes, they will split up 2–2. If the middle

[7] One can show that if the excitation of SF after 1 activation spreading step is e_j, the final activity will be $\frac{1}{1-\alpha}\left(e_j - \frac{\alpha}{1+\alpha(N-1)}\Sigma_i e_i\right)/\mathit{spreadingRate}$, where α is the inhibitory weight (i.e. 0.1) and N is the number of SF nodes (i.e. 10).

category has three SF nodes, any of three things can happen: the nodes of the middle category split 2–1; the nodes split 2–1 but the second middle category borrows a node from its neighbour; or the category with four nodes splits 2–2.

If, conversely, a learner with four categories moves to a place with three, she will merge two categories, typically the two in the middle.

If all three nodes of the second category (4–5–8) die, the remaining seven nodes will divide themselves up between the three categories. If the whole of the higher-frequency third of AudF dies, its three nodes will be recruited by the first and second categories.

We conclude that the network has a high degree of plasticity, adapting itself to changes in the environment as well as to changes in its own structure.[8]

Comparison with earlier models 5.11

A potential early stage of categorical perception, the *perceptual magnet effect* (Kuhl 1991), has been modelled with neural nets before by Guenther and Gjaja (1996). This work had four aspects that make it difficult to use their model for our purposes. First, the learning rule was instar, which does not work for auditory dispersion (Section 6). Second, the inputs were only four AudF nodes, with a formant value unrealistically represented by the activity levels of two AudF nodes rather than by an array of nodes as here. Third, the state of SF was selected less realistically (i.e. more "manually") than here, namely by setting all activities that did not exceed a certain threshold to zero (rather than by mutual inhibition). Fourth, the magnet effect was established by computing a "population vector" based on a computation of auditory distance; in our case, a "warped" AudF can be directly computed by clamping an AudF to an incoming CoG value, then computing the output SF, then clamping the SF at this output, then unclamping AudF and having activity spread back to it from SF; this reflection works correctly thanks to the bidirectionality of the connections, which Guenther and Gjaja could not implement.

[8] In human learners, plasticity may well decay with age, so that adaptation to changing environments slows down as the child grows older. Modelling this lies outside the scope of the present paper.

Some aspects of our model are shared with the TRACE model by McClelland and Elman (1986), the most notable being upward activation spreading. The critical difference, however, between our model and TRACE, in the light of Section 5, is that TRACE featured "local" (i.e. non-distributed) representations and therefore had to work with pre-given categories. Even if TRACE had come with a learning algorithm, which McClelland and Elman did not provide, TRACE thus could not have handled the main objective of Section 5, which is the emergence of categories. A phenomenon that TRACE did successfully account for is the *Ganong effect* (Ganong III 1980), by which low-level perception (for us, the mapping from AudF to SF) is influenced by top–down information (for us, from Morphemes down to SF) about the existence of lexical items (for us, Morphemes). Simulating the Ganong effect in our model would equally require a third level of representation that encodes meaning and feeds information back to SF. This should be possible, because the BiPhon model of Figure 2 provides the required level of representation (UF and/or Morphemes), and adding such a level above the SF of the present section would provide the required feedback as a result of the bidirectionality of the connections. While the simulations presented here do not address the exact effects of the top–down feedback from a third level (though see Chládková 2014 for showing that top–down effects do happen when we add a third layer), they do illustrate that our model satisfies another prerequisite for the Ganong effect, namely ambiguity at the middle level. In TRACE, the Ganong effect is critically dependent on ambiguity at the middle level of representation, which is resolved by top–down activation from existing lexical items. In our simulations, ambiguity at SF occurs in the mixed excitations visible in Figure 16 (e.g. the 4th picture from the bottom in the left column, and the 4th picture from the top in the right column) and Figure 17 (around nodes 11 and 20), which occur in response to AudF input that lies in a distributional valley (i.e. near a boundary between two ambient categories). These two-level simulations in our model thus lay the foundation for a full simulation of the Ganong effect, which has to be postponed to future work that investigates three levels of representation.

AUDITORY DISPERSION 6

Auditory dispersion is a phenomenon in sound change whereby the auditory correlates of phonological elements become optimally distributed along one or more auditory dimensions. The emergence of auditory dispersion over the generations was handled successfully in BiPhon-OT (Boersma and Hamann 2008). In this section, we test whether BiPhon-NN is equally capable of doing the job.

Existing work on auditory dispersion 6.1

Languages tend to maximize the auditory contrast between elements in their phonological inventories (e.g. Passy 1890; von der Gabelentz 1901; de Groot 1931; Martinet 1960). In a single auditory dimension, languages favour symmetric inventories whose members lie at equal distances along the auditory continuum, often with a preference for the centre, as in Figure 23. If we take as an example of an auditory continuum the voice onset time (VOT) in bilabial plosives, Estonian would be an example of a language with a single category, namely /p/, which is realized with zero VOT (Figure 23a), Swedish exemplifies a language with two categories, namely /b/, realized with negative VOT, and /pʰ/, realized with positive VOT (Figure 23b), and Thai serves to illustrate that a language can have the three categories /b/, /p/ and /pʰ/ (Figure 23c).

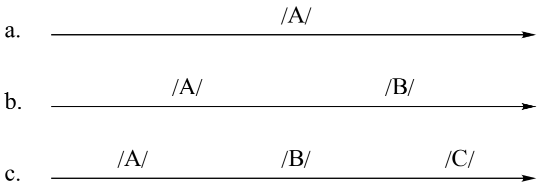


Figure 23:
Typically dispersed phonological inventories

Inventories as in Figure 23 are *optimally dispersed* in the sense that they strike a perfect balance between perceptual clarity and articulatory ease (Lindblom 1986; ten Bosch 1991; Boersma 1998). Practically speaking, optimal auditory dispersion entails that the categories are sufficiently auditorily distinct to minimize confusion in the listener, and that this distinctivity does not come at too large an articulatory cost for the speaker.

Boersma and Hamann (2008) formalize auditory dispersion within BiPhon-OT as the result of an interaction between cue constraints, whose ranking is a result of optimizing the learner's prelexical perception during acquisition, and articulatory constraints, which aim for articulatory ease. When re-using the perception-optimized cue constraint ranking in production (phonetic implementation), the dispersion effect automatically emerges. With computer simulations, Boersma and Hamann show that optimally dispersed systems are diachronically stable, and that poorly dispersed systems evolve into stable systems within a small number of generations. The BiPhon-OT account is devoid of teleological devices, such as the explicit auditory-distance maximization by Liljencrants and Lindblom (1972), ten Bosch (1991) or Schwartz *et al.* (1997), or such as the OT dispersion constraints proposed by Flemming (1995/2002: MINDIST), Kirchner (1998/2001: DISP), and Padgett (2003: SPACE), whose sole purpose was to preclude categories from approaching each other; in fact, the listener does not have to compute auditory distances at all, as was still the case with some less-teleological methods, such as the agent-based simulations by de Boer (1999) and Oudeyer (2006), and such as Wedel's (2006) exemplar-based account.

6.2 *A neural network for auditory dispersion*

We will try to replicate Boersma and Hamann's results with BiPhon-NN. We propose that after the unsupervised bottom-up creation of categories of Section 5, the learner creates a lexicon of phonological word forms (at UF), which is capable of "supervising" perceptual learning. That is, once the learner has established a lexicon, the lexicon can provide top–down information, in effect telling the network what phonological category to expect, or what phonological category it should have perceived. To this end, we consider the neural network in Figure 24, which just as the one we used in Section 5.7 has three layers: the phonological surface form (SF), the auditory-phonetic form (AudF), and the articulatory-phonetic form (ArtF).

The network has nine SF nodes for a distributed representation of the categories. As was approximately the case throughout Section 5,

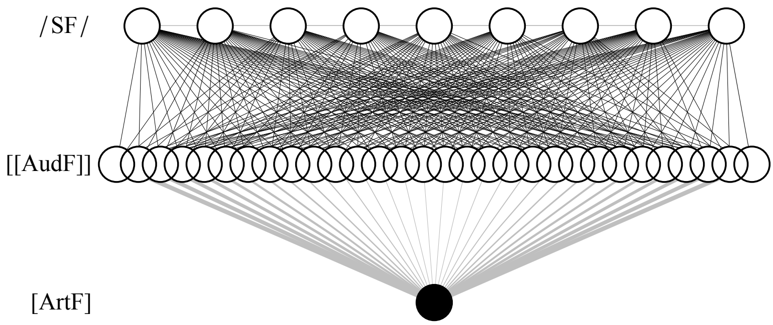


Figure 24:
The initial state of the neural network

each discrete phonological category is represented by three SF nodes: category 1 corresponds to SF nodes 1, 4, and 7, category 2 to nodes 2, 5, and 8, and category 3 to nodes 3, 6, and 9. As before, there are inhibitory connections within SF.

The AudF layer again represents the CoG dimension, sampled again in 30 steps. Each AudF node is connected to all nine SF nodes by excitatory cue connections (drawn in black) whose initial weights have random values between 0 and 0.1. Each AudF node is also connected to the ArtF node by an inhibitory articulatory connection (drawn in light grey); these connections have the same values as in 5.7: they are stronger (i.e. drawn thicker) at the edges of the AudF layer, to represent the idea that the production of a peripheral value requires more articulatory effort than the production of a central value.

Learning to perceive 6.3

The simulated learner will have to establish the appropriate cue connection weights of the ambient language through a process of perceptual learning. Before the learning process begins, we create the initial language: for every category, we determine a normal distribution of input probabilities along the auditory continuum. In each learning step, a combination of a category and an auditory value is selected at random; if a value has a high input probability given the selected category, it is more likely to be drawn. We pair each auditory value with a category because we want the learning process to be supervised by information from “above”, i.e. from the lexicon at and/or above

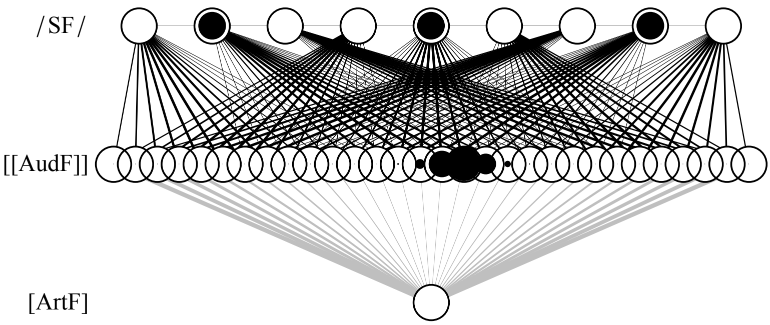


Figure 25: The neural network after 50,000 learning steps

UF and perhaps also from the phonology of the UF-to-SF mapping: somewhat artificially, we assume that the learner's lexicon is already in place, i.e. that she knows what category she should have perceived. We switch on the selected AudF nodes as well as the selected category nodes at SF; subsequently, all AudF and SF nodes are clamped, and the weights of the cue connections are updated with the inoutstar rule (Section 4.7).

After 50,000 tokens (learning rate = 0.01) from a language with input peaks as in Figure 13, i.e. at 16.667% of the auditory continuum (category 1), at 50% (category 2) and at 83.333% (category 3), the network from Figure 24 comes to look as Figure 25. The left third of the AudF layer is more strongly connected to SF nodes 1, 4 and 7 than to other SF nodes, so the network has learned that low auditory values are most likely to be intended as category 1; likewise, mid auditory values connect to category 2, and high auditory values to category 3, as the language environment dictated.

6.4 *Production: the articulatory effect*

The network is bidirectional, so it uses the same connections in production as in perception. Figure 26 shows how the network of Figure 25, which has been trained only to perceive, handles production. To see how a category is produced, we switched on its three SF nodes (activity 0.8), as shown by filled disks in the figure, while switching off the other six SF nodes (activity zero), as shown by empty disks; all nine SF nodes are clamped at these values, as shown by solid circles. Now the ArtF node also comes into play, clamped at an activity of 1.0,

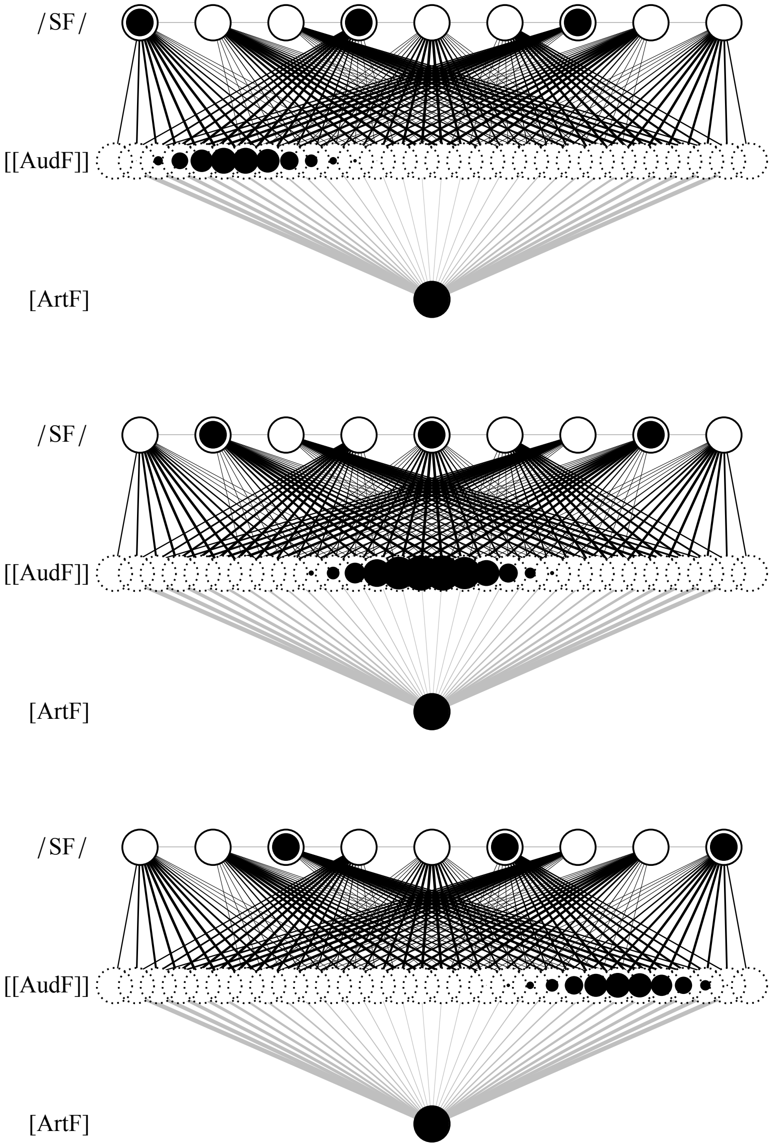


Figure 26: Output activities for the three categories (peaks in input distribution as in Figure 13)

constraining the activities at the unclamped AudF layer. After activity spreads from SF and ArtF to AudF for 500 time steps, Figure 26 shows the resulting activities on the AudF layer (as usual, negative activities are clipped at zero) in the production of each of the three categories. The strongest activities in Figure 26 are between nodes 6 and 7 (i.e. at 5.5/29 = 19.0% of the continuum), between nodes 15 and 16 (50%), and at node 24 (23/29 = 79.3% of the continuum).

The locations of the strongest activities are important concepts. According to Section 2.5, we can regard these locations as the most probable auditory forms realized in production. When we look at their values, we see that they are different from what the learner has heard in her environment. The learner has shifted category 1 by 19.0% − 16.7% = 2.3% toward the centre of the continuum, when compared to her language environment, and she shifted category 3 toward the centre by 83.3% − 79.3% = 4.0%. These values of 2.3% and 4.0% are typical: if we repeat the experiment, we see that learners will on average shift the two outside categories by 3% toward the centre of the continuum.

It is clear where this shift comes from. As in 5.7, it comes from the articulatory constraints: auditory values around 19% and 79% are just somewhat easier to produce than values around 17% and 83%, so the learner's cue constraints might prefer values around 19% and 79%, but her articulatory constraints move the values away from this effortful periphery.

6.5 *Production: the prototype effect*

The question is: will learners always shift the categories toward the centre? That would be bad for the future of the language, because a sequence of learners would ultimately make all categories pile up in the very centre of the continuum, where they merge into one.

Fortunately, near the centre of the continuum a different effect counteracts the articulatory effect. Figure 27 shows a network that has learned 50,000 times from a "confusing" language where the distributions of the three categories have peaks at 40%, 50% and 60%. The strongest cue constraints now connect the three categories at SF to much more central auditory values than in Figure 25. The production, however, works as in Figure 28. The strongest activities are at

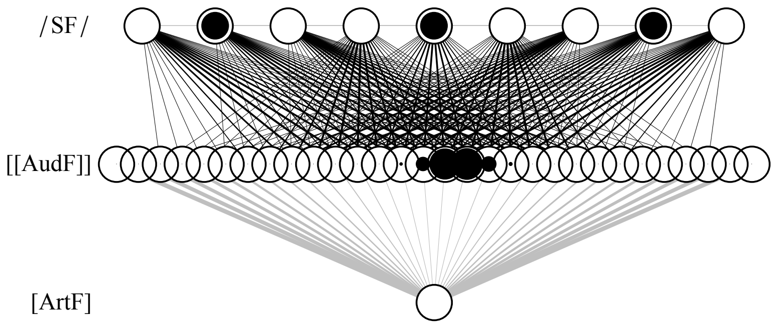


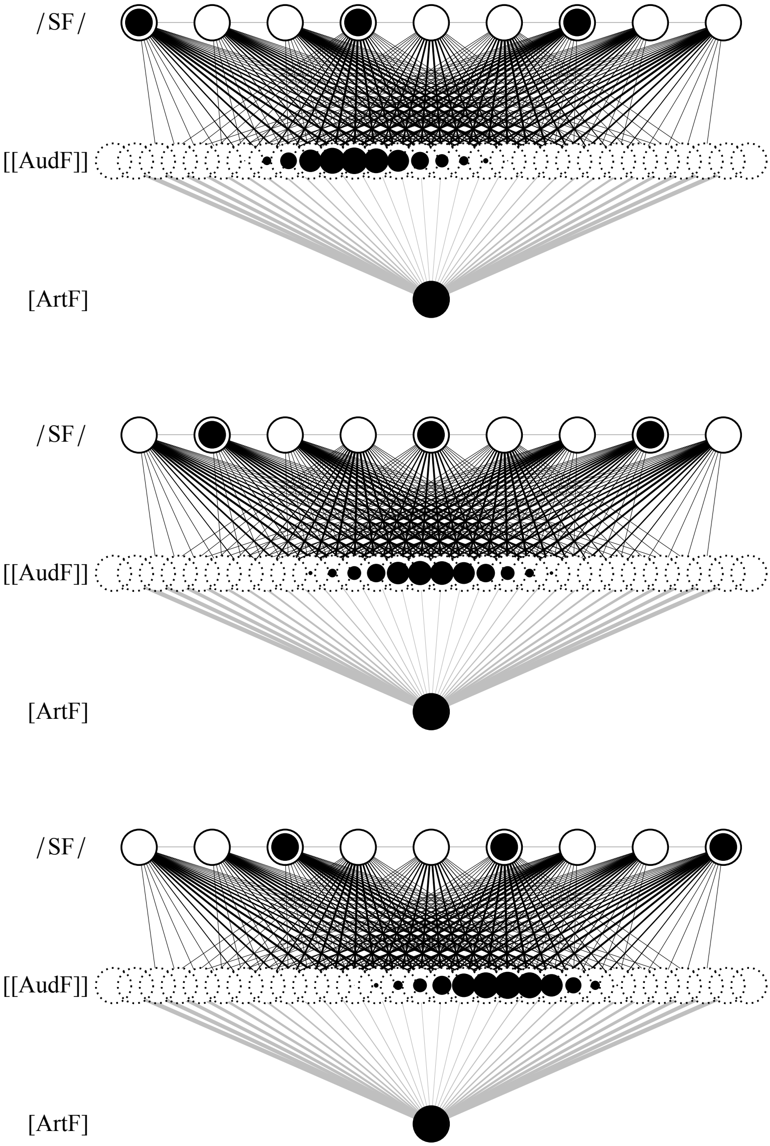
Figure 27: The neural network after 50,000 pieces of confusing data

node 12 (i.e. at 11/29 = 37.9% of the continuum), between nodes 15 and 16 (at 50%), and at node 19.3 or so (18.3/29 = 63.1% of the continuum). The two outside categories, therefore, have shifted 40%−37.9% = 2.1% and 63.1%−60% = 3.1% toward the periphery of the continuum.

What happened here? The outstar part of the learning algorithm makes stronger connections between AudF and SF if the probability of that SF given that AudF is greater; in fact, the weight moves asymptotically toward the conditional probability of that SF given that AudF. Now, a more peripheral AudF value (say, at 30% of the continuum) is more likely to have been intended as category 1 than a more central AudF value (say, at 40% of the continuum), because around 40% of the continuum we are in a region where the distribution of category 1 overlaps with the distribution of category 2. As a result, the connection between an AudF of 30% and category 1 will be stronger than the connection between an AudF of 40% and category 1. As a result, the production of category 1 will favour an AudF of 30% over an AudF of 40%. This result replicates the observation that listeners choose more peripheral tokens as prototypical than they produce themselves (Johnson *et al.* 1993; explained with BiPhon-OT by Boersma 2006). The inoutstar algorithm employed here does not exhibit this "prototype effect" (Boersma and Hamann 2008) as strongly as the outstar algorithm, but it employs it enough to shift the category by several percent.

Summing up, then, categories whose centres lie near the periphery of the auditory continuum will tend to shift toward the centre, whereas categories that overlap with other categories will tend to

Figure 28: Output activities for the three categories (peaks in input distribution as in Figure 27)



move away from those other categories. Over the generations, an equilibrium will appear where all categories are approximately equally spaced around the centre of the continuum; the distances between the category centres will not depend on where they were in the first generation.

Our simulations show, then, that BiPhon-NN, just as BiPhon-OT, is capable of replicating the emergence of optimal dispersion in phonological inventories. If the network learns the appropriate weights of the cue constraints in comprehension and then "produces" sound using the same connections, any input distribution will evolve into a stable system within a number of generations. It is thus crucial that the neural network is symmetric, as it is in other models that involve both sensory input and production (Kohonen 1984; Wedel 2007).

For more details on the properties of the neural network and learning procedure used here, and for simulations of other inventories, we refer the reader to Seinhorst (2012), who also subjects to closer scrutiny the difference between outstar and inoutstar learning in modelling auditory dispersion.

DISCUSSION 7

One and the same network, with a single learning rule, namely "inoutstar" learning, has turned out to be able to handle both category creation (in a slightly brittle manner) and auditory dispersion (very robustly). While the instar rule would have worked fine for category creation (as Guenther and Gjaja 1996 have shown), and the outstar rule works fine for the emergence of auditory dispersion (as shown by Seinhorst 2012), only the inoutstar rule, which is a combination of the instar and outstar rules, works for both.

On top of the two foci of the present paper (category creation and auditory dispersion), the BiPhon-NN model replicates several realistic behavioural effects, with minimal assumptions and devices. Although the model does not represent or compute auditory distance (as earlier models of both category creation and dispersion did do; see Section 5.11 and Section 6.1), realistic effects of auditory vicinity emerge both in category creation and in dispersion, because the model automatically learns the correlation between adjacent auditory values in the input (Section 5.5). Although the model employs identical knowledge in the comprehension and production directions, asymmetries between comprehension and production do arise in the realistic prototype effect (Section 6.5). And although the more comprehensive model of Figure 1 includes levels that are non-adjacent and therefore seems to disallow nonlocal interactions, it can achieve realistic effects of interactivity across multiple levels, because activity spreads simultaneously top–down and bottom–up (as in the TRACE model; see Section 5.11); an example of this in Section 5.7 and Section 6.4 is the interactive effect of the "later" articulation on the "earlier" mapping from SF to AudF in production.

On the downside, the model cannot really represent more than one segment yet, and we have not attempted to supply the networks with time-varying input at the auditory or surface level. As a result, no phonological structure beyond single categories can be represented yet in the distributed versions of the network, and interesting issues involving time-varying perception or production, such as contextual cue weighting, dynamic sensorimotor knowledge, or coarticulation, could not be studied yet. Once these sequence restrictions are overcome at all levels of representation, important questions that can be answered are whether anything similar to the within-level restrictions of Figure 1 emerges in these networks, and whether anything emerges that is similar to the many hierarchical structures that have been proposed in the literature. Such issues point toward a large-scale programme for future research.

8 CONCLUSION

The BiPhon-NN model is seen to handle some phenomena that psycholinguists and speech researchers have found in the lab and that have never been modelled within a single framework before. Also, the BiPhon-NN model is biologically one step more plausible than an OT model. One of the main missing areas involves strictly phonological phenomena, which will require the model to come to represent sequential or hierarchical structures at the level of the phonological surface form.

REFERENCES

David H. ACKLEY, Geoffrey E. HINTON, and Terrence J. SEJNOWSKI (1985), A learning algorithm for Boltzmann machines, *Cognitive Science*, 9:147–169.

Diana APOUSSIDOU (2007), *The learnability of metrical phonology*, Ph.D. thesis, University of Amsterdam.

Titia BENDERS (2013), *Nature's distributional-learning experiment: infants' input, infants' perception, and computational modeling*, Ph.D. thesis, University of Amsterdam.

Iris BERENT, Tracy LENNERTZ, Paul SMOLENSKY, and Vered VAKNIN-NUSBAUM (2009), Listeners' knowledge of phonological universals: evidence from nasal clusters, *Phonology*, 26:75–108.

Catherine BEST (1995), A direct realist view of cross-language speech perception, in Winifred STRANGE, editor, *Speech perception and linguistic experience: theoretical and methodological issues*, pp. 171–203, York Press, Baltimore.

Paul BOERSMA (1997), How we learn variation, optionality, and probability, *Proceedings of the Institute of Phonetic Sciences (University of Amsterdam)*, 21:43–58.

Paul BOERSMA (1998), *Functional phonology: formalizing the interactions between articulatory and perceptual drives*, Ph.D. thesis, University of Amsterdam.

Paul BOERSMA (2000), The OCP in the perception grammar, Rutgers Optimality Archive 435, `http://roa.rutgers.edu`.

Paul BOERSMA (2006), Prototypicality judgments as inverted perception, in Gisbert FANSELOW, Caroline FÉRY, Ralf VOGEL, and Matthias SCHLESEWSKY, editors, *Gradience in grammar: generative perspectives*, pp. 167–184, Oxford University Press, Oxford.

Paul BOERSMA (2007), Some listener-oriented accounts of *h*-aspiré in French, *Lingua*, 117:1989–2054.

Paul BOERSMA (2009), Cue constraints and their interactions in phonological perception and production, in Paul BOERSMA and Silke HAMANN, editors, *Phonology in perception*, pp. 55–110, Mouton De Gruyter, Berlin.

Paul BOERSMA (2011), A programme for bidirectional phonology and phonetics and their acquisition and evolution, in Anton BENZ and Jason MATTAUSCH, editors, *Bidirectional Optimality Theory*, pp. 33–72, John Benjamins, Amsterdam.

Paul BOERSMA (2012), Modelling phonological category learning, in Abigail C. COHN, Cécile FOUGERON, and Marie K. HUFFMAN, editors, *The Oxford handbook of laboratory phonology*, pp. 207–218, Oxford University Press, New York.

Paul BOERSMA, Paola ESCUDERO, and Rachel HAYES (2003), Learning abstract phonological from auditory phonetic categories: an integrated model for the acquisition of language-specific sound categories, in Maria-Josep SOLÉ, Daniel RECASENS, and Joaquin ROMERO, editors, *Proceedings of the 15th International Congress of Phonetic Sciences, Barcelona*, pp. 1013–1016, Futurgraphic, Barcelona.

Paul BOERSMA and Silke HAMANN (2008), The evolution of auditory dispersion in bidirectional constraint grammars, *Phonology*, 25:217–270.

Paul BOERSMA and Silke HAMANN (2009), Loanword adaptation as first-language phonological perception, in Andrea CALABRESE and W. Leo WETZELS, editors, *Loanword phonology*, pp. 11–58, John Benjamins, Amsterdam.

Paul BOERSMA and Bruce HAYES (2001), Empirical tests of the Gradual Learning Algorithm, *Linguistic Inquiry*, 32:45–86.

Paul BOERSMA and Jan-Willem VAN LEUSSEN (2017), Efficient evaluation and learning in multi-level parallel constraint grammars, *Linguistic Inquiry*, 48:349–388.

Ludwig BOLTZMANN (1868), Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten, *Wiener Berichte*, 58:517–560.

György BUZSÁKI and Kenji MIZUSEKI (2014), The log-dynamic brain: how skewed distributions affect network operations, *Nature Reviews Neuroscience*, 15:264–278.

Kateřina CHLÁDKOVÁ (2014), *Finding phonological features in perception*, Ph.D. thesis, University of Amsterdam.

Anne CUTLER, Jacques MEHLER, Dennis NORRIS, and Juan SEGUI (1987), Phoneme identification and the lexicon, *Cognitive Psychology*, 19:141–177.

Bart DE BOER (1999), *Self-organisation in vowel systems*, Ph.D. thesis, Vrije Universiteit Brussel.

Willem DE GROOT (1931), Phonologie und Phonetik als Funktionswissenschaften, *Travaux du Cercle Linguistique de Prague*, 4:146–147.

Paola ESCUDERO and Paul BOERSMA (2004), Bridging the gap between L2 speech perception research and phonological theory, *Studies in Second Language Acquisition*, 26:551–585.

Edward FLEMMING (1995/2002), *Auditory representations in phonology*, Ph.D. thesis, University of California at Los Angeles, published in 2002 by Routledge (New York/London).

Carol A. FOWLER (1986), An event approach to the study of speech perception from a direct-realist perspective, *Journal of Phonetics*, 14:3–28.

William F. GANONG III (1980), Phonetic categorization in auditory word perception, *Journal of Experimental Psychology: Human Perception and Performance*, 6:110–125.

Stephen D. GOLDINGER (1996), Words and voices: episodic traces in spoken word identification and recognition memory, *Journal of Experimental Psychology: Learning, Memory and Cognition*, 5:1166–1183.

Stephen GROSSBERG (1969), Embedding fields: a theory of learning with physiological implications, *Journal of Mathematical Psychology*, 6:209–239.

Stephen GROSSBERG (1976), Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors, *Biological Cybernetics*, 23:121–134.

Stephen GROSSBERG (1987), Competitive learning: from interactive activation to adaptive resonance, *Cognitive Science*, 11:23–63.

Frank H. GUENTHER and Marin N. GJAJA (1996), The perceptual magnet effect as an emergent property of neural map formation, *Journal of the Acoustical Society of America*, 100:1111–1121.

Richard H.R. HAHNLOSER, Rahul SARPESHKAR, Misha A. MAHOWALD, Rodney J. DOUGLAS, and H. Sebastian SEUNG (2000), Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, *Nature*, 405:947–951.

Mark HALE and Charles REISS (2000), “Substance abuse” and “dysfunctionalism”: current trends in phonology, *Linguistic Inquiry*, 31:157–169.

Donald O. HEBB (1949), *The organization of behavior*, Wiley, New York.

John HOPFIELD (1982), Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–2558.

Keith JOHNSON, Edward FLEMMING, and Richard WRIGHT (1993), The hyperspace effect: phonetic targets are hyperarticulated, *Language*, 69:505–528.

Paul KIPARSKY (1982), From cyclic phonology to lexical phonology, in Harry VAN DER HULST and Norval SMITH, editors, *The structure of phonological representations*, volume I, pp. 131–175, Foris, Dordrecht.

Robert KIRCHNER (1998/2001), *An effort-based approach to consonant lenition*, Ph.D. thesis, University of California at Los Angeles, published in 2001 by Routledge (New York/London).

Teuvo KOHONEN (1984), *Self-organization and associative memory*, Springer, Berlin.

John K. KRUSCHKE (1992), ALCOVE: An exemplar-based connectionist model of category learning, *Psychological Review*, 99:22–44.

Patricia K. KUHL (1991), Human adults and human infants show a “perceptual magnetic effect” for the prototypes of speech categories, monkeys do not, *Perception and Psychophysics*, 50:93–107.

Pierre-Simon LAPLACE (1812), *Théorie analytique des probabilités*, Veuve Courcier, Paris.

Willem LEVELT, Ardi ROELOFS, and Antje MEYER (1999), A theory of lexical access in speech production, *Behavioral and Brain Sciences*, 22:1–75.

Alvin LIBERMAN and Ignatius MATTINGLY (1985), The motor theory of speech perception revised, *Cognition*, 21:1–36.

Alvin M. LIBERMAN, Katherine Safford HARRIS, Howard S. HOFFMAN, and Belver C. GRIFFITH (1957), The discrimination of speech sounds within and across phoneme boundaries, *Journal of Experimental Psychology*, 54:358–368.

Johan LILJENCRANTS and Björn LINDBLOM (1972), Numerical simulation of vowel quality systems: the role of perceptual contrast, *Language*, 48:839–862.

Björn LINDBLOM (1986), Phonetic universals in vowel systems, in John OHALA and Jeri JAEGER, editors, *Experimental phonology*, pp. 13–44, Academic Press, Orlando.

Rafael LORENTE DE NÓ (1938), Cerebral cortex: architecture, intracortical connections, motor projections, in J.F. FULTON, editor, *Physiology of the nervous system*, pp. 291–327, Oxford University Press, London.

R. Duncan LUCE (1959), *Individual choice behavior: a theoretical analysis*, Wiley, New York.

André MARTINET (1960), *Éléments de linguistique générale*, Armand Colin, Paris.

Jessica MAYE, Daniel J. WEISS, and Richard N. ASLIN (2008), Statistical phonetic learning in infants: facilitation and feature generalization, *Developmental Science*, 11:122–134.

James L. MCCLELLAND and Jeffrey L. ELMAN (1986), The TRACE model of speech perception, *Cognitive Psychology*, 18:1–86.

Bob MCMURRAY, Jessica S. HORST, Joseph C. TOSCANO, and Larissa K. SAMUELSON (2009), Towards an integration of connectionist learning and dynamical systems processing: case studies in speech and lexical development, in John P. SPENCER, Michael S.C. THOMAS, and James L. MCCLELLAND, editors, *Toward a unified theory of development: connectionism and dynamic systems theory re-considered*, pp. 218–249, Oxford University Press, New York.

Dennis NORRIS (1994), Shortlist: a connectionist model of continuous speech recognition, *Cognition*, 52:189–234.

Dennis NORRIS, James M. MCQUEEN, and Anne CUTLER (2000), Merging information in speech recognition: feedback is never necessary, *Behavioral and Brain Sciences*, 23:299–370.

Erkki OJA (1982), A simplified neuron model as a principal component analyzer, *Journal of Mathematical Biology*, 15:267–273.

Pierre-Yves OUDEYER (2006), *Self-organization in the evolution of speech*, Oxford University Press, Oxford.

Randall C. O'REILLY (1996), *The Leabra model of neural interactions and learning in the neocortex*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Jaye PADGETT (2003), Contrast and post-velar fronting in Russian, *Natural Language and Linguistic Theory*, 21:39–87.

Paul PASSY (1890), *Étude sur les changements phonétiques et leur caractères généraux*, Firmin-Didot, Paris.

Joe PATER (2004), Bridging the gap between receptive and productive development with minimally violable constraints, in René KAGER, Joe PATER, and Wim ZONNEVELD, editors, *Constraints in phonological acquisition*, pp. 219–244, Cambridge University Press, Cambridge.

Donald H. PERKEL and Theodore H. BULLOCK (1969), Neural coding, *Neurosciences Research Symposium Summaries*, 3:405–527.

Janet PIERREHUMBERT (2001), Exemplar dynamics: word frequency, lenition and contrast, in Joan BYBEE and Paul HOPPER, editors, *Frequency and the emergence of linguistic structure*, pp. 137–157, John Benjamins, Amsterdam.

Alan PRINCE and Paul SMOLENSKY (1993/2004), Optimality Theory: constraint interaction in generative grammar, Technical Report TR-2, Rutgers University Center for Cognitive Science, published in 2004 by Blackwell (Malden, MA/Oxford).

Nathaniel ROCHESTER, John H. HOLLAND, Luther H. HAIBT, and William L. DUDA (1956), Tests on a cell assembly theory of the action of the brain, using a large digital computer, *IRE Transactions on Information Theory*, 2:80–93.

Frank ROSENBLATT (1958), The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, 65:386–408.

David E. RUMELHART and David ZIPSER (1985), Feature discovery by competitive learning, *Cognitive Science*, 9:75–112.

Jean-Luc SCHWARTZ, Louis-Jean BOË, Nathalie VALLÉE, and Christian ABRY (1997), The dispersion–focalization theory of vowel systems, *Journal of Phonetics*, 25:255–286.

Klaas SEINHORST (2012), *The evolution of auditory dispersion in symmetric neural nets*, Master's thesis, University of Amsterdam.

Klaas SEINHORST, Paul BOERSMA, and Silke HAMANN (2019), Iterated distributional and lexicon-driven learning in a symmetric neural network explains the emergence of features and dispersion, in Sasha CALHOUN, Paola

ESCUDERO, Marija TABAIN, and Paul WARREN, editors, *Proceedings of the 19th International Congress of Phonetic Sciences, Melbourne*, pp. 1135–1138, Australasian Speech Science and Technology Association Inc., Canberra.

Paul SMOLENSKY (1996), On the comprehension/production dilemma in child language, *Linguistic Inquiry*, 27:720–731.

Louis TEN BOSCH (1991), *On the structure of vowel systems: aspects of an extended vowel model using effort and contrast*, Ph.D. thesis, University of Amsterdam.

Sophie TER SCHURE, Caroline JUNGE, and Paul BOERSMA (2016), Semantics guide infants' vowel learning: computational and experimental evidence, *Infant Behavior and Development*, 43:44–57.

Bruce TESAR (1997), An iterative strategy for learning metrical stress in Optimality Theory, in Elizabeth HUGHES, Mary HUGHES, and Annabel GREENHILL, editors, *Proceedings of the 21st Annual Boston University Conference on Language Development*, pp. 615–626, Cascadilla, Somerville, MA.

Bruce TESAR and Paul SMOLENSKY (1998), Learnability in Optimality Theory, *Linguistic Inquiry*, 29:229–268.

Bruce TESAR and Paul SMOLENSKY (2000), *Learnability in Optimality Theory*, MIT Press, Cambridge, MA.

Nikolai TRUBETZKOY (1939), *Grundzüge der Phonologie*, Travaux du Cercle Linguistique de Prague 7.

Nicolaas VAN WIJK (1936), Positieve en negatieve opmerkingen over de definitie van het phoneem, *Nieuwe Taalgids*, 30:311–326.

Georg VON DER GABELENTZ (1901), *Die Sprachwissenschaft: ihre Aufgaben, Methoden und bisherigen Ergebnisse*, Tauchnitz, Leipzig.

Karin WANROOIJ (2015), *Distributional learning of vowel categories in infants and adults*, Ph.D. thesis, University of Amsterdam.

Richard M. WARREN and Roslyn P. WARREN (1970), Auditory illusions and confusions, *Scientific American*, 223:30–37.

Andrew WEDEL (2004), *Self-organization and categorical behavior in phonology*, Ph.D. thesis, University of California at Santa Cruz.

Andrew WEDEL (2006), Exemplar models, evolution and language change, *The Linguistic Review*, 23:247–274.

Andrew WEDEL (2007), Feedback and regularity in the lexicon, *Phonology*, 24:147–185.

David WEENINK (2006), *Speaker-adaptive vowel identification*, Ph.D. thesis, University of Amsterdam.

Bernard WIDROW and Marcian E. HOFF (1960), Adaptive switching circuits, in *IRE WESCON Convention Record, part 4*, pp. 96–104, IRE, New York, reprinted in James E. ANDERSON and Edward ROSENFELD, editors (1988), *Neurocomputing*, pp. 126–134, MIT Press, Cambridge, MA.

Paul Boersma
0000-0003-4328-3840
paul.boersma@uva.nl

Klaas Seinhorst
0000-0002-9716-0742
seinhorst@uva.nl

Amsterdam Center
for Language and Communication
University of Amsterdam
Amsterdam, The Netherlands

Titia Benders
0000-0003-0143-2182
titia.benders@mq.edu.au

Department of Linguistics
and Centre for Language Sciences
Macquarie University
Sydney, Australia

Paul Boersma, Titia Benders, and Klaas Seinhorst (2020), *Neural network models for phonology and phonetics*, Journal of Language Modelling, 8(1):103–177
https://dx.doi.org/10.15398/jlm.v8i1.224