Neural network models for phonology and phonetics[1]

Paul Boersma, Titia Benders, Klaas Seinhorst

24 July 2012

Abstract. This paper 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. Specifically, the model is 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 solves the hitherto unsolved problem of how symbolic-looking discrete language behavior 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, contain 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 behavioral data that have been assembled, namely 1) the generalizations that phonologists have found within and across languages, 2) the phenomena that psycholinguists and speech researchers have found 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 behavioral 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.

1.1. *A model of phonological and phonetic representations and knowledge*

If the model consists of levels of representation, it may look like Fig.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.

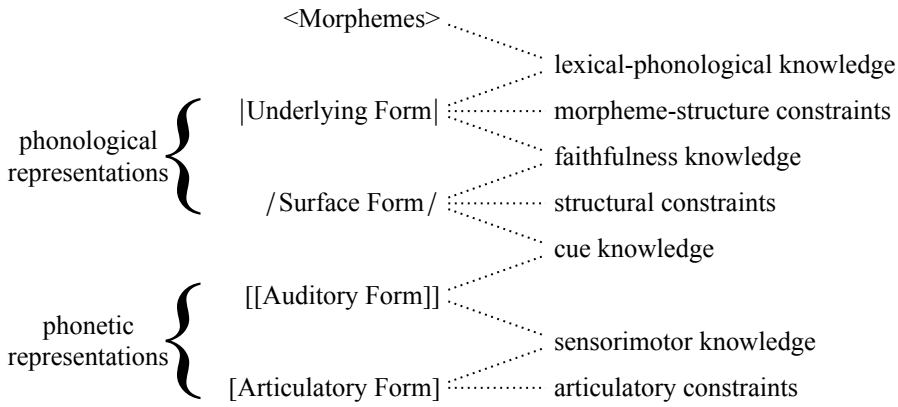

Fig. 1. Levels of representation and stored knowledge in a model of phonology and phonetics.

The five levels in Fig. 1 are a simplified combination of what phonologists have been proposing in models of phonological production (e.g. refxx structuralism, Kiparsky 1982) and what psycholinguists have been proposing in models of comprehension (e.g. Cutler 19xx) and production (e.g. Levelt, Roelofs and Meyer 1999). These specific five levels, and the special way in which they are connected in Fig. 1, were established by Boersma (1998, 2007) and Apoussidou (2007). In numerous papers, Boersma and co-workers 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 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 Fig. 1, such relationships exist between adjacent levels only, so that the language user has knowledge about sensorimotor, cue, phonological and lexical relationships. The language user also has knowledge about restrictions within levels: the articulatory, structural and morpheme-structure constraints. In OT, such knowledge is represented as a grammar consisting of ranked constraints; in NN models, such 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 behavior 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 Fig. 1. Globally, the path from AudF to Morphemes following the upward arrows in Fig. 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.

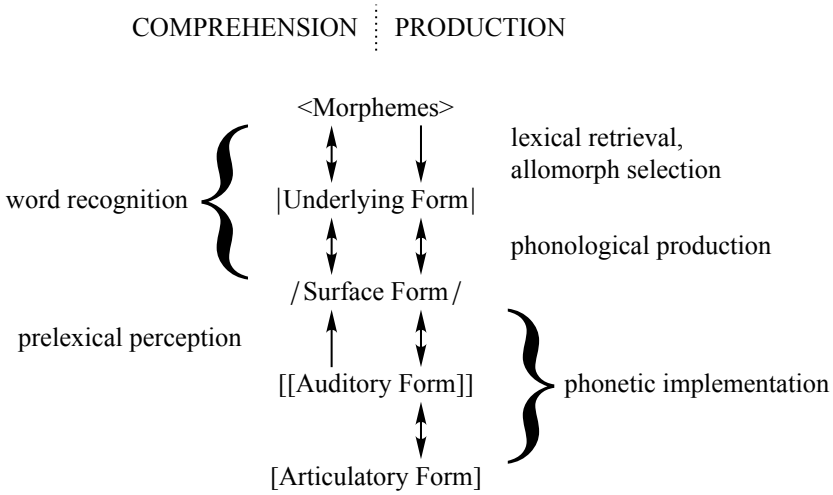

Fig. 2. Processes in a comprehensive model of phonology and phonetics.

The partial processes and their acquisition have been modeled in various frameworks. Phonologists have been modeling phonological production within OT since Prince and Smolensky (1993), and its acquisition since Tesar & Smolensky (1998). The acquisition of prelexical perception was modeled with neural networks such as the perceptron by refxx, and within BiPhon-OT by Boersma (1997) and Escudero and Boersma (2004). [xx Norris (1994) geeft met Shortlist een NN implementation van SF>UF mapping] The present paper in section 5 handles the *perceptual magnet effect*, i.e. perceptual warping as an early stage of category creation in the AudF-to-SF mapping, which was observed in the lab by Kuhl (1992). The emergence of this effect was modeled before with neural networks by Guenther and Gjaja (1996) and with BiPhon-OT by Boersma, Escudero and Hayes (2003).

The way in which the language user's knowledge is represented in Fig. 1 suggest that the same knowledge is used for both directions of processing in Fig. 2, i.e. for comprehending and producing speech. For 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 & 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 you to perceive a loud high-frequency noise as /s/ forces you to implement the phoneme /s/ as a sound with a loud high-frequency noise. In section 6 we handle the phenomenon of *auditory dispersion*, i.e. the evolution of optimal distances at AudF between the members of phoneme inventories at SF (refxx). This was modeled 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 §6.

Thus, the perceptual magnet effect and auditory dispersion were both modeled before, although never within the same framework.

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 §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 Fig. 1 or 2. We discuss these reasons now, with the goal of finding candidate comprehensive modeling 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 behavior, 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 Figs. 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 effort constraints. An explanation of loanword adaptation, therefore, requires all arrows in Fig. 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 slower than the initial adaptation of a loanword, but is also much more central to linguistic theory and experimentation. The search, therefore, is for a single comprehensive framework.

1.3.2. *Distant interactions* The arrows in Fig. 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" levels of representation to "earlier" levels in processing has been identified experimentally and theoretically in several locations, 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 found in listeners by Ganong (1980), and modeled 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 (Warner and Warner 1970). In production, allomorph selection is sometimes determined by 'later' considerations at SF, such as that between |vjø| and |vjɛj| 'old-MASC' in French. 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 modeled by Boersma (1998, 2007). Also, cue knowledge and articulatory constraints must interact in the phonetic implementation process.

As a result of these examples of *interactive* processing, most of the arrows in Fig. 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 (McQueen, Cutler and Norris 2000 deny the influence of the lexicon on prelexical perception, and Hale and Reiss 20xx 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 §1.1 through §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 modeling category creation (Boersma 1998: ch.8; Boersma, Escudero and Hayes 2003) the BiPhon model shares with NN category creation models (refxx, Guenther and Gjaja 1996) and noncomputational emergentist work (refxx, Blevins 2004) 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 have to stop there, because linguistic modeling in e.g. OT requires that categories are discrete. This discrepancy between the gradiency of category creation that is needed in an emergentist model, and the discreteness of categories that is needed to do OT phonology, means 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 one has 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.

Exemplar theory (refxx) might do better with respect to the transition between continuous and discrete (massive storage of single events leads to observed continuous knowledge), but 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).

Which leaves neural network modeling. If Fig. 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. The pattern of activity of these nodes forms the current representation at that level. The processes of Fig. 2 can be regarded as the spreading of activity between and within levels; the knowledge in Fig. 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 behavior 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 behavior. 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 modeling 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.

Let's proceed to looking at the ingredients of our linguistic NN model.

2. Nodes, connections, weights and activities

2.1. *A toy example: phonological production*

We introduce artificial neural networks by looking at a traditional toy example of phonological production. Using terms that are familiar from both the neural network literature (refxx) and OT (Prince and Smolensky 1993: xx), the Underlying Form is the *input* of this mapping and the Surface Form is the *output*.

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 /ampa/ on the surface 70% of the time, and 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 we know (Stochastic) OT can represent (e.g. Boersma 2008), 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 C in 70% of the cases and as 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 behavior.

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 Fig. 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.

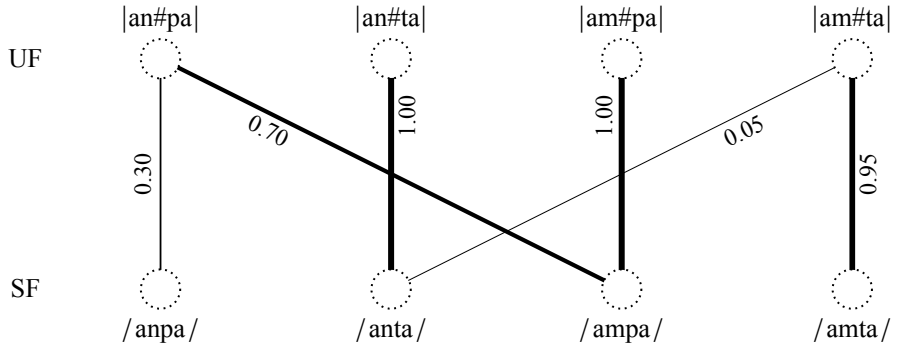

Fig. 3. A network that performs phonological production.

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 §4.

In Fig. 3, each node is represent by 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 (refxx, Hopfield 1982), and they guarantee the bidirectionality (§1.2) of the BiPhon model, thus providing the desired dispersion effect in §6. The present paper can do with, and indeed crucially employs, bidirectional connections; if in future modeling this simplification turns out to be untenable, it can then be dispensed with.

In NN modeling, 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 Fig. 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 neuron fires, a neuron with which it has a thick (strong) synapse will be influenced stronger than a neuron with which it has a thinner (weaker) synapse. 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).

To be more precise about the numbers: we deliberately chose the weight of each UF–SF connection in Fig. 3 to be equal to the probability that the connected UF node is realized as the connected SF node in our toy language. As a further example of this, an underlying "homorganic" |an#ta| has 100% chance of being 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 Fig. 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 (§2.2), additive excitation of the output nodes (§2.3), a linear excitation-to-activity function (§2.4), and a linear activity-to-probability function (§2.5). We illustrate these concepts with Fig. 4, which shows the production of underlying |an#pa|.

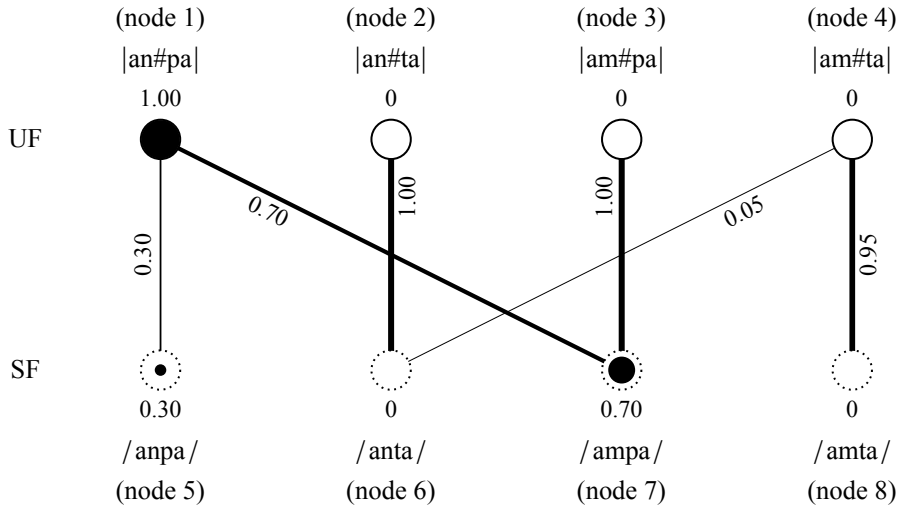

Fig. 4. The production of underlying |an#pa|.

2.2. *Activity 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 Fig. 4: by painting the whole node in black, and (in this figure only) 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, xx spikes per second (refxx); a node with an activity of 0 can be seen as a neuron (or group of neurons) with a minimum firing frequency (say, xx spikes per second; refxx).

The circles for the UF nodes in Fig. 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*.

2.3. *Excitation of the output nodes*

When an input node is activated, as node |an#pa| is in Fig. 4, the information about its activity will spread towards the nodes with which it is connected: the activity will *excite* every connected node to some extent. For instance, in Fig. 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 therefore 1.00·0.70 = 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 0·1.00 = 0. Node |an#pa| excites /ampa/ by an amount 0 (the activity of |an#pa|) times 0 (the weight of the connection from |an#pa| to /ampa/), which is 0·0 = 0, and so does |am#ta|.

Biologically, these four excitations can be regarded as "post-synaptic 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 (refxx). 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

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

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 (refxx) 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:

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

As a result, 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 stay between 0 and 1 (which seems reasonable, given the biological interpretation of these limiting values as the minimum and maximum possible firing

frequency), one could simply clip the activity between those values, maintaining linearity of all activities between 0 and 1:

$$a_j = \max\left(0, \min\left(e_j, 1\right)\right) \qquad (\text{for } j = 5,8) \tag{3}$$

or one could apply a "top-sigmoid" clipping, which is linear for small excitations and goes to 1 smoothly for large excitations:

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

In the end, combining the assumption of additive excitation (the contributions from the four underlying forms are added up) and the assumption of a linear excitation-to-activity function (the activity of an output node is a linear function of 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.

2.5. *Probabilistic interpretation of the activity of the output nodes*

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 behavior, we will have to provide a behavioral interpretation of the result in Fig. 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 behavior 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, Rumelhart and Zipser 1985). Here, however, this option would throw away the /anpa/ candidate entirely, and such nonstochastic behavior 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 §5.xx).

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 an output candidate is its activity, scaled by the sum of all output activities:

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

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%.

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 behavior, the relative activities have apparently determined the relative probabilities of the behavior.

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

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

where T is called the *temperature*. The simpler linear relation of (5), however, will suffice for the present paper.

2.6. *Bidirectionality violated?*

The network of Fig. 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 Fig. 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 (refxx), this should shift the preference of a listener towards |an#pa|, but in the network of Fig. 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 §5. Understanding such a network, however, requires understanding how the activities of equation (1) come about in processing (§3), and how the weights in Fig. 3 come about in learning (§4).

3. Activity spreading

In the example of §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 §2, the excitation in the output nodes 5 through 8 starts at zero, and is incremented at every time step (say, every millisecond) with an amount Δe_j given by

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

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

To see that (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 towards 0.70.

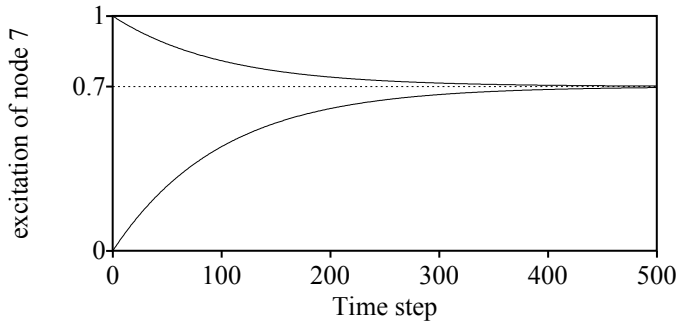

Fig. 5. The time path of the excitation (and activity) of node /ampa/.
Bottom curve: starting from 0. Top curve: starting from 1.00.

One can predict the end result 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 Fig. 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 towards a constant final state, as in Fig. 5, is general for symmetric networks (refxx). 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 towards such a stable final state.

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

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

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. Learning in a linguistic network

The representations and processes discussed in §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 Fig. 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 explain how this can happen in our artificial networks.

4.1. *Learning the toy language from UF–SF pairs*

Suppose we have the toy language of §2.1, with the coronal bias of §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 start with the network of Fig. 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 Fig. 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 Fig. 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. We will now discuss six ways to do that.

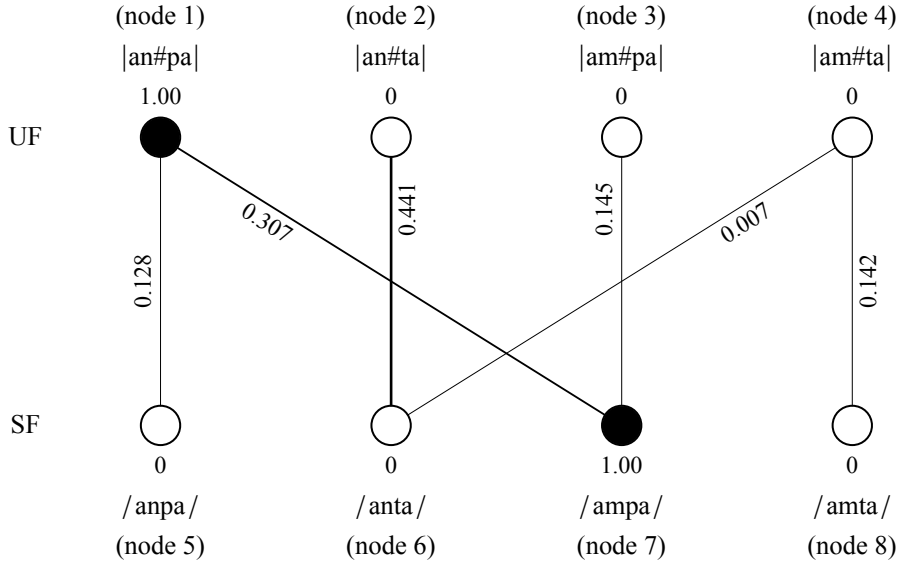

Fig. 6. Supervised two-level learning: all nodes are clamped, and only one node is on in UF as well as SF.

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 formula:

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

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_1 and a_7 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.

If this 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 Fig. 7.

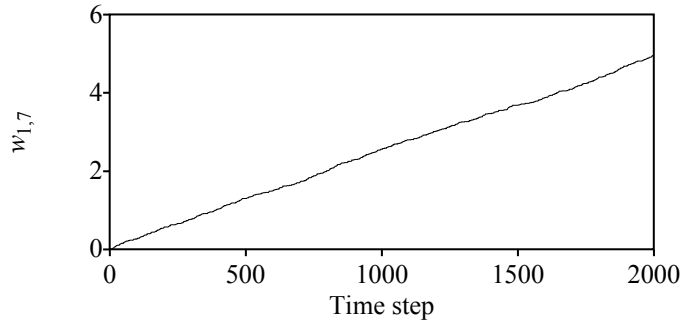

Fig. 7. The development of a weight in pure Hebbian learning: linear and without bounds.

We see that w_{ij} increases linearly with time, and goes on to do that without bounds. It has been known from the beginning of neural network modeling that this "pure Hebbian learning" exhibits this pathological behavior (refxx). This is named after Hebb (1949), who proposed that a synaptic strength increases when two neurons fire together, but he did not propose formula (9). Various devices have been proposed in the literature to keep w_{ij} within bounds.

4.3. *Clipped linear learning*

A brute-force method (refxx) 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 is known to have the tendency of ultimately pushing most weights towards 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 any scenario from §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. This situation is not good for stochastic decision-making, where we want probabilities to move gradually from 0 to 1 or the reverse.

4.4. *Leaky learning*

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

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

The weights now start to rise exactly as in Fig. 6, but after some time they start to rise more slowly, growing exponentially towards an equilibrium in very much the same way as in Fig. 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 Fig. 8.

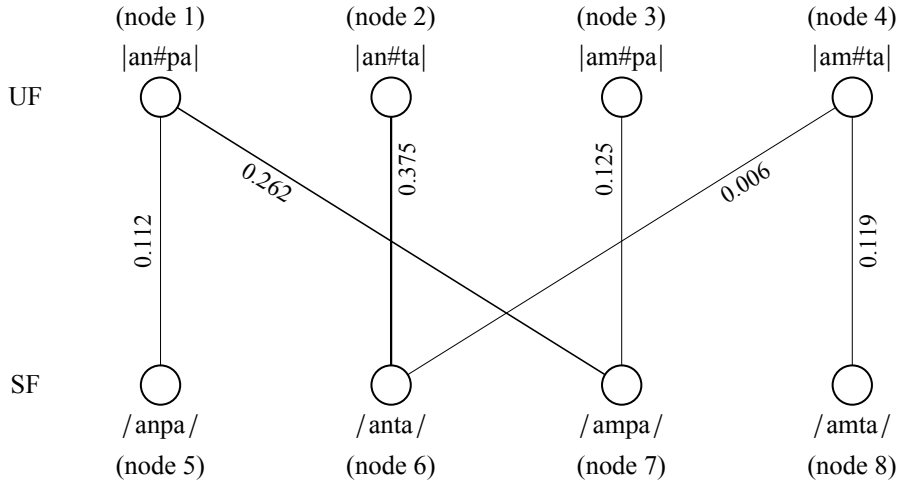

Fig. 8. The average end state of leaky learning in the language environment of §4.1.

Each weight in Fig. 8 is exactly the probability of the relevant UF–SF pair as mentioned in §4.1; the sum of all the weights in the figure is 1. We could have predicted this result from equation (10) by realizing that in the equilibrium situation the expected weight change $\langle \Delta w_{ij} \rangle$ is 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 §4.1, is zero. Equation (10) then tells us that the expectation value $\langle a_i a_j - w_{ij} \rangle$ will then move towards zero, so that the weight w_{ij} will ultimately go toward the correlation between a_i and a_j:

$$w_{ij} \quad \rightarrow \quad \langle a_i a_j \rangle \tag{11}$$

Thus, 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 $\langle a_i a_j \rangle$ = 0.2625·1·0 + 0.1125·1·1 + 0.625·0·0 + 0·0·1 = 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:

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

Such pure correlation learning looks nice but has a disadvantage. Relatively rare inputs will lead to weak connections: |am#pa| has a three times weaker connection in Fig. 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 large difference in the activities at SF, which means that further on in processing the rare UF counts less *much* less heavily than the more frequent UF. A learning rule that focuses on reliability rather than frequency alone may fare better, and is certainly closer to how OT handles rare inputs. 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/).

4.5. *Outstar learning*

One way to take reliability into account is the outstar learning rule (refxx):

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

For our toy language, this leads to the weights in Fig. 9.

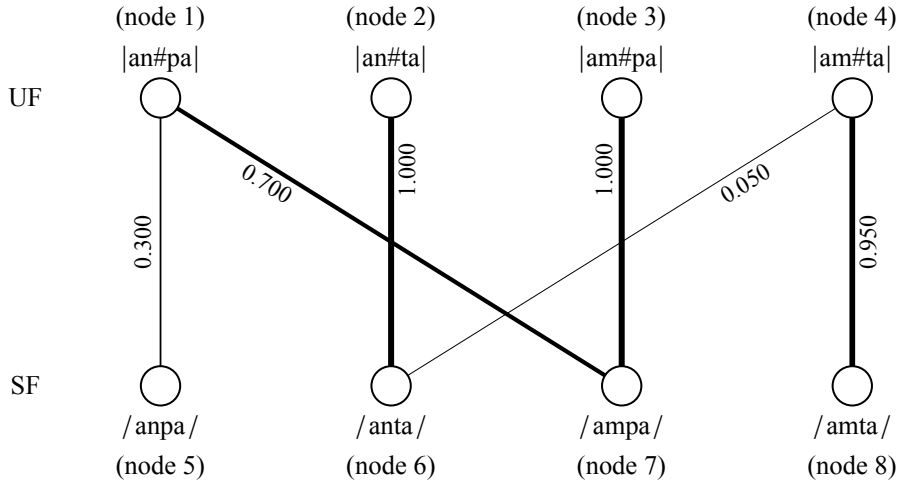

Fig. 9. The average end state of outstar learning in the language environment of §4.1.

The weights turn out to become the conditional probabilities of SF given UF, so it exhibits the probability-matching behavior that we wanted; the sum of the weights going out from each UF node is 1. This could have been predicted by realizing that in the equilibrium situation $0 = \left\langle a_i a_j - a_i w_{ij} \right\rangle = \left\langle a_i a_j \right\rangle - \left\langle a_i \right\rangle w_{ij}$, so the weights must go to

$$w_{ij} \quad \rightarrow \quad \frac{\left\langle a_i a_j \right\rangle}{\left\langle a_i \right\rangle} \tag{14}$$

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:

$$w_{ij} \quad \rightarrow \quad \frac{\mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_i = 1\right)} = \mathrm{P}\left(a_j = 1 \,\middle|\, a_i = 1 \right) \tag{15}$$

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. What is lost now is all dependency of SF activity on the frequency of the input, for which there is evidence in the literature (refxx); a way to have both reliability and frequency influences is to have a combination of (10) and (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 Fig. 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.

Outstar learning is close to the *delta rule* of supervised learning algorithms (refxx), 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 §2:

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

This, together with the property of probabilities conditional to the input, makes this algorithm a good candidate for our use in modeling the auditory dispersion effect in §6.

4.6. *Instar learning*

To take the specificity of SF into account, we can apply the instar learning rule (Grossberg 1976, Rumelhart & Zipser 1985), which is the outstar learning rule in the opposite direction of processing:

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

For our toy language, this leads to the weights in Fig. 10.

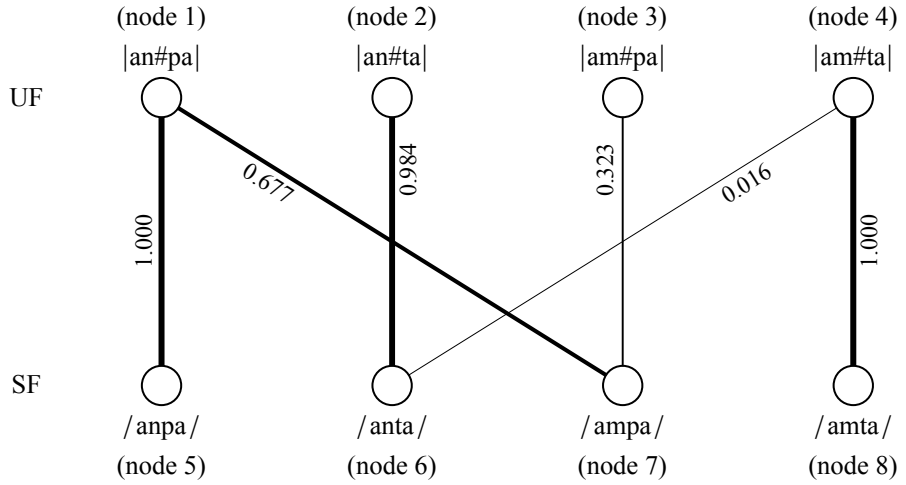

Fig. 10. The average end state of instar learning in the language environment of §4.1.

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 equilibrium situation

$$w_{ij} \rightarrow \frac{\langle a_i a_j \rangle}{\langle a_j \rangle} \tag{18}$$

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:

$$w_{ij} \rightarrow \frac{\mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_j = 1\right)} = \mathrm{P}\left(a_i = 1 \,\middle|\, a_j = 1 \right) \tag{19}$$

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).

Instar learning is known from work on competitive learning (Grossberg 1976, Rumelhart & Zipser 1985). We can therefore expect that it works well on problems of category creation, including our modeling of the perceptual magnet effect of §5.

4.7. *Inoutstar learning*

To model category creation we seem to need unsupervised instar learning, and to model auditory dispersion we seem to need supervised outstar learning. 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 the perceptual magnet effect and auditory dispersion with a single learning algorithm, perhaps a compromise between instar and outstar. We call this the "inoutstar" learning rule:

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

For our toy language, this leads to the weights in Fig. 11.

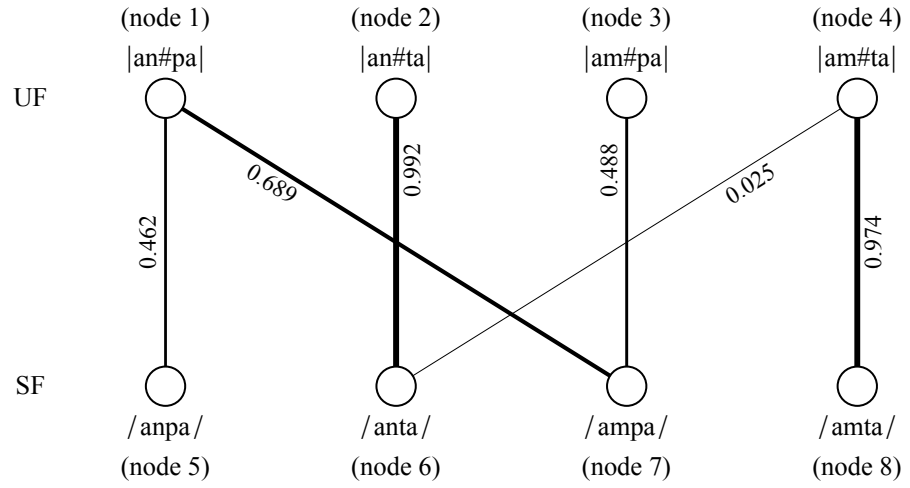

Fig. 11. The average end state of inoutstar learning in the language environment of §4.1.

Each weight turns out to become the harmonic mean of the weights of Figs. 9 and 10. In the equilibrium situation

$$w_{ij} \quad \rightarrow \quad \frac{2\left\langle a_i a_j \right\rangle}{\left\langle a_i + a_j \right\rangle} \tag{21}$$

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:

$$w_{ij} \quad \rightarrow \quad \frac{2\,\mathrm{P}\left(a_i = 1 \wedge a_j = 1\right)}{\mathrm{P}\left(a_i = 1\right)\mathrm{P}\left(a_j = 1\right)} = \frac{2\,\mathrm{P}\left(\, a_i = 1 \,\middle|\, a_j = 1 \,\right)\mathrm{P}\left(\, a_j = 1 \,\middle|\, a_i = 1 \,\right)}{\mathrm{P}\left(\, a_i = 1 \,\middle|\, a_j = 1 \,\right) + \mathrm{P}\left(\, a_j = 1 \,\middle|\, a_i = 1 \,\right)} \tag{22}$$

Inoutstar learning 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 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} in the beginning of learning: equation (22) shows that inoutstar learning causes the weights to *become* symmetric.

4.8. *Conclusion*

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

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

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 §5 and §6 we investigate the suitability of this algorithm for two hitherto separately modeled phenomena.

5. Phonological category creation

In this section we investigate category creation at the phonology-phonetics interface, i.e. the emergence of discrete behavior at SF on the basis of continuous input at AudF.

5.1. *A network for category emergence*

Figure 12 shows the network that should perform the task of category creation.

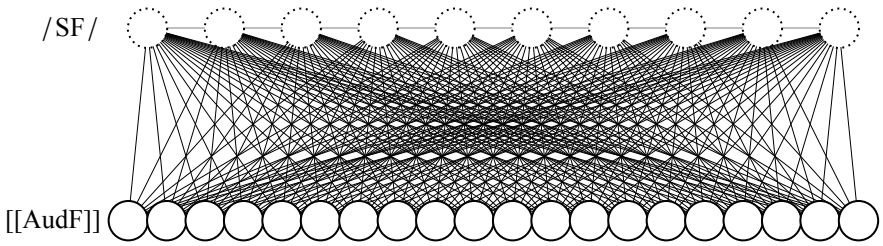

Fig. 12. A network for category creation, with continuous sound coming in at clamped AudF and discrete behavior emerging at unclamped SF.

Figure 12 contains 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 output. The figure displays the initial state of the network, with small random weights between AudF and SF.

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 simplify matters here by representing the spectrum with only 20 nodes, running from low frequencies at the left to high frequencies at the right. We simplify even more by allowing the incoming sound to activate only one small region of AudF; this simplification means that the continuum can represent a unitary spectral continuum, and for this we choose the first formant (F1).

The Surface Form in Fig. 12 will come to represent a phonological feature, namely phonological vowel height, because that is the feature that has F1 as its main auditory correlate. SF consists of 10 nodes, which should be more than enough to represent any number of vowel height categories that can occur in languages (there can be anywhere between two and six). In the initial state of Fig. 12, the network cannot classify incoming

auditory forms into stable categories yet, because the weights between AudF and SF are still small and random: an incoming sound (i.e. a local activity peak in AudF) will just produce a random and small pattern on SF.

AudF and SF are fully connected: there are 200 connections between AudF and SF, one for each pair of AudF node and SF node. Initially, these weights are random: uniformly distributed between 0 and 0.1. The weights will change during learning. We thereby hope to model the emergence of categorical behavior in SF.

There are also 45 connections within SF: one for each pair of SF nodes. These connections have fixed negative weights of -0.2 (shown in light gray) in order to make sure that the nodes react to different incoming patterns, a mechanism we borrow from competitive learning models (Grossberg 1976, Rumelhart and Zipser 1985). These weights do not change during learning.

5.2. *An input distribution for vowel height*

The simplest way to model category creation is to feed the network F1 values at AudF, without telling the network the associated higher levels of representation, such as meaning. For instance, the language environment of our virtual learner could consist of three vowels, namely /i/, /e/ and /a/, as in a language with three vowel heights such as Spanish. We assume that the F1 of each of these three vowels is distributed according to a Gaussian distribution, as in the three dotted curves in Fig. 13. The learner, however, does not know that there are three curves; she is only confronted with the sum of the distributions, namely the total F1 distribution shown as the solid curve in Fig. 13. In this example, the distance between the peaks is one third of range of the continuum, and the standard deviation of each peak is one third of that distance; as a result, the valleys are rather shallow, namely approximately 65% of the height of the peaks. It is on the basis of these peaks and valleys alone that the learner will have to figure out that there are three categories.

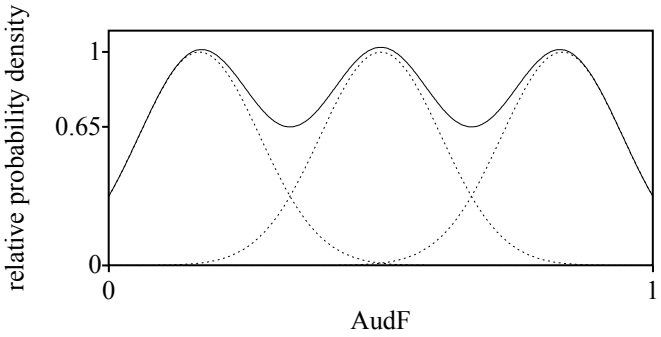

Fig. 13. An F1 distribution in a language with three vowel heights.

5.3. *Unsupervised teaching of the distribution*

We cannot feed the network the distribution of Fig. 13 at one stroke. Instead, a single F1 value is drawn from the distribution randomly each time the learner's language environment produces an utterance. The learner receives this F1 value as an activity at AudF, then spreads this activity to SF, then updates the weights on the basis of the activities at AudF and SF. Amazingly, this procedure does lead to the emergence of categorical behavior at SF after 10,000 or so incoming F1 values, as we will now show.

Whenever a single F1 value drawn from the summed distribution in Fig. 13 is applied to AudF, this produces an activity pattern at AudF of the form shown in Fig. 14. The F1 value is turned into a node number between 1 and 20, but not rounded. In Fig. 14, an F1 value is turned into node number 15.3. The nodes in the vicinity of location 15.3 are then activated according to a Gaussian shape with a standard deviation of 1/21 of the continuum. This activates node 15 the strongest (at a distance of 0.3), then node 16 (distance 0.7), then node 14 (1.2), then 17, then 13, and so on; the activities of nodes further away are too weak to be visible in the figure.

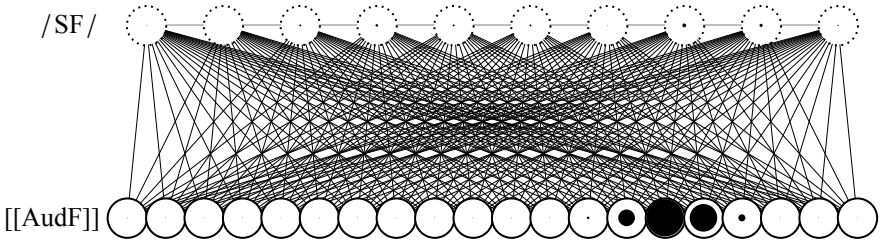

Fig. 14. Applying an input.

After the input is applied to AudF, the AudF nodes in Fig. 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 the dotted edges), 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, and soon also SF, excite the nodes at SF according to equation (8), with a spreading rate of 0.01, an excitation leak of 1, and a shunting factor of 0; basically, this is equation (7), except that the summation is over all AudF and SF nodes and the equation applies to all SF nodes. The computation of activity from excitation simply follows equation (2), i.e. the activities do not clip. Spreading goes on in this way for 100 time steps. The result is that after the first time step, the activities at SF start to influence SF, and ultimately the whole network would move towards 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 (20) applied to all 200 connections between AudF and SF. There is only one learning step per incoming F1 value.

5.4. *Result after learning: three categories have emerged*

After 20,000 incoming F1 values, the weights of the network have become those in Fig. 15.

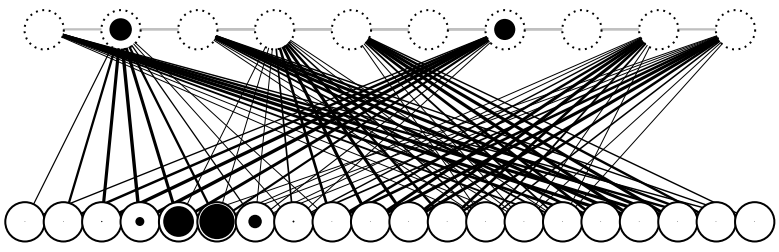

Fig. 15. A categorizing network.

At SF, nodes 2 and 7 (i.e. the two that are on in the figure) have become associated to low ([i]-like) F1 values, nodes 4, 9 and 10 to intermediate ([e]-like) F1 values, and nodes 1, 3 and 5 to high ([a]-like) F1 values. Nodes 6 and 8 have disconnected from AudF.

This situation is categorical behavior. When pacing a local activity pattern through AudF, the output at SF comes to favor exactly three stable types of patterns of activity, as can be seen in Fig. 16. For AudF nodes 1 through 6 (the first six pictures on the left), only SF nodes 2 and 7 switch on. Since activity patterns are the brain's way of representing behavior, the stable 2–7 pattern at SF represents a stable type of behavior at (and above) SF. This is what it means to be a category in neural-network terms: a stable type of behavior. We can call this category the "2–7" category when looking at this learner's SF; it replicates the /i/ category of the language of the parents.

The degree of the activities within a category at SF is not always the same: the activities are much higher for AudF nodes 3 and 4 (where the peak of the first category is located, as can be seen in Fig. 13) than for AudF nodes 1 and 6. Thus, the first category is much stronger activated for the relatively common sounds around nodes 3 and 4 than for the less frequent sounds around nodes 1 and 6. Since category goodness is therefore represented at SF, it will not be surprising if the learner turns out to be able to report differences in category goodness for the various sounds that she classifies as the same category.

There are two more stable types of behavior, i.e. two more categories. For AudF nodes 8 through 13, the SF pattern (which we could label as /e/) is 4–9–10, and for AudF nodes 15 through 20 the SF pattern (/a/) is 1–3–5. At the category boundaries, a mixed type of behavior appears. For AudF node 7, SF shows a combination of the 2–7 category and the 4–9–10 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 /i/ and /e/ both move on towards UF, activating in the lexicon words with underlying |i| as well as words with underlying |e|. Since AudF node 7 can indeed represent two categories from the language environment, such uncertainty is adaptive and appropriate. Something similar happens at AudF node 14: its reflex at SF is a mixture of the 4–9–10 (/e/) and 1–3–5 (/a/) categories.

We conclude that there come to be three types of stable behavior 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 behavior increases from nothing (the random behavior that the network of Fig. 14 would exhibit) to almost perfect (the behavior of the network in Figs. 15 and 16). Thus, **categoryhood is gradient** in this model: nearly categorical behavior emerges before strictly categorical behavior does.

A difference between the final network of Fig. 15 and the networks we discussed in sections 2 through 4, is that the network of Fig. 15 no longer represents a phonological category as a single node, but that it 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 of such a network is redundancy: if a couple of AudF nodes die, and one SF node dies, the network will still perform its classification task quite well: in Fig. 15, 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.

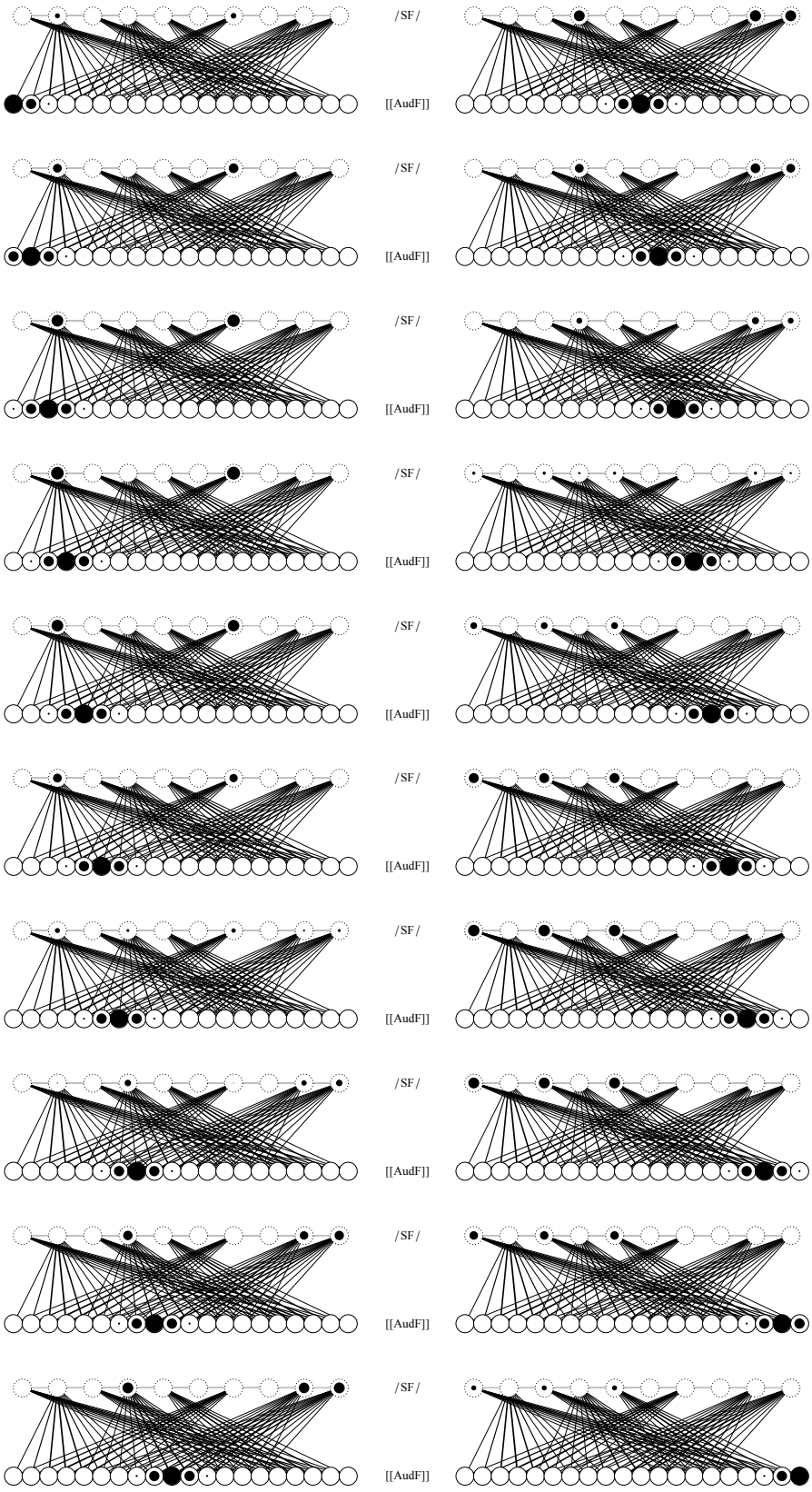

Fig. 16. Pacing through the Auditory Form yields three types of patterns in the Surface Form.

5.5. *Replicating experimental data: categorical perception and the perceptual magnet effect*

It is known that listeners can discriminate easier two auditory forms that map to different phonological categories than two auditory forms that map to the same category (Liberman et al. 1957). The network of Fig. 15 can replicate this behavior, 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 Fig. 15, we first compute the average absolute difference between the activities of the SF nodes in the first two pictures in Fig. 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 17 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 centers. This *discrimination curve* illustrates the categorical perception effect.

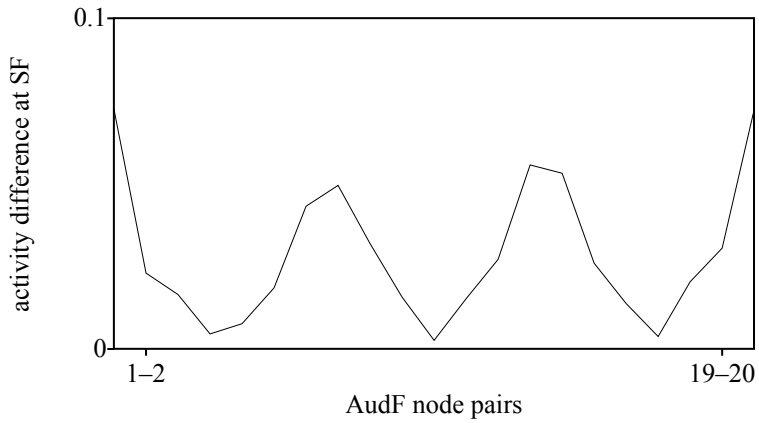

Fig. 17. The discrimination curve. The peaks at the edges represent the difference between nodes 1 and 20.

A potential early stage of categorical perception, the perceptual magnet effect (Kuhl 1991), has been modeled 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 (§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 activities at SF were selected less realistically than here, namely by setting all activities that did not exceed a certain threshold to zero. 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 F1 value, then computing the output SF, then clamping the SF at this output, then unclamping AudF and have 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.

6. Auditory dispersion

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.

6.1. *Existing work on auditory dispersion*

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 favor symmetric inventories whose members lie at equal distances along the auditory continuum, often with a preference for the center, as in Fig. 18.

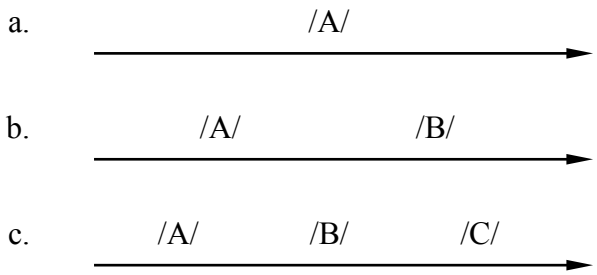

Fig. 18. Typically dispersed phonological inventories.

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 (Fig. 18a); e.g. Swedish has two categories, namely /b/, realized with negative VOT, and /p^h/, realized with positive VOT (Fig. 18b); and Thai has the three categories /b/, /p/ and /p^h/ (Fig. 18c).

Inventories as in Fig. 18 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 dispersion constraints proposed by Flemming (1995/2002: MINDIST), Kirchner (1998: DISP), and Padgett (2003: SPACE), whose sole purpose was to preclude categories from approaching each other; nor does the listener have to compute auditory distances, as in 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. Figure 19 shows a neural network like the one we used in section 5, but now with three layers: the phonological surface form (SF), the auditory-phonetic form (AudF), and the articulatory-phonetic form (ArtF).

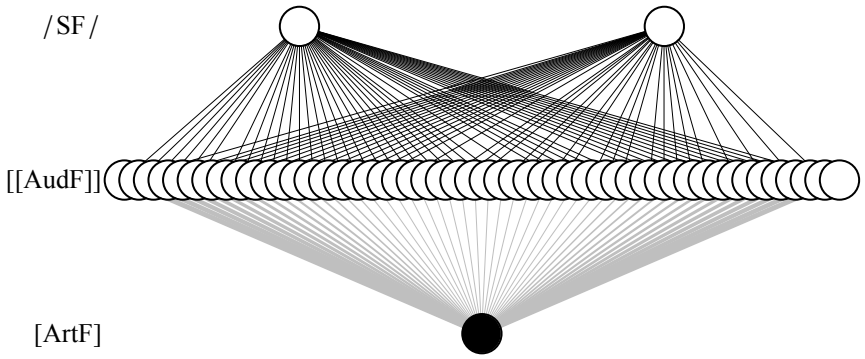

Fig. 19. The initial state of the neural network.

The network has two SF nodes, each corresponding to a discrete phonological category. At the SF layer, a local rather than a distributed representation (§5) was chosen in order to simplify the modeling of the production direction. Since the two or three nodes that represent a category in §5 came to have the same connection weights to their auditory forms, so that they were always activated equally strongly, we can simplify the production model by using one SF node per category, without loss of generality.

The AudF layer again represents the F1 dimension, sampled this time in 50 steps for more precision (for the simulations in section 5 the number of steps does not matter). Each AudF node is connected to both 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 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.

Note that the connections between the AudF and ArtF layers are not sensorimotor connections; the speaker is assumed to have perfect sensorimotor knowledge. In a more extensive network layout there would be exactly as many ArtF nodes (articulatory-phonetic representations) as there are AudF nodes, and the articulatory connections would be connections within the ArtF layer (§1); the present network layout, with just one ArtF node, is chosen in order to make the interaction of auditory and articulatory factors more perspicuous.

6.3. *Learning to perceive*

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, the initial language is created: for every category, a normal distribution of input probabilities along the auditory continuum is computed. 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. Combinations of

categories and auditory values are chosen because the learning process is supervised by the lexicon: somewhat artificially, it is assumed that the learner's lexicon is already in place, i.e. she knows what category she should have perceived. The selected AudF node is switched on, as is the selected category node; subsequently, all AudF and SF nodes are clamped, and the weights of the cue connections are updated with the outstar rule (§4.5), which replicates the OT behavior in optimizing mappings (we will investigate inoutstar later). After 50,000 tokens (learning rate = 0.01) from a language with input peaks at 35% (left category) and 65% (right category) of the auditory continuum, the network from Fig. 19 comes to look as Fig. 20:

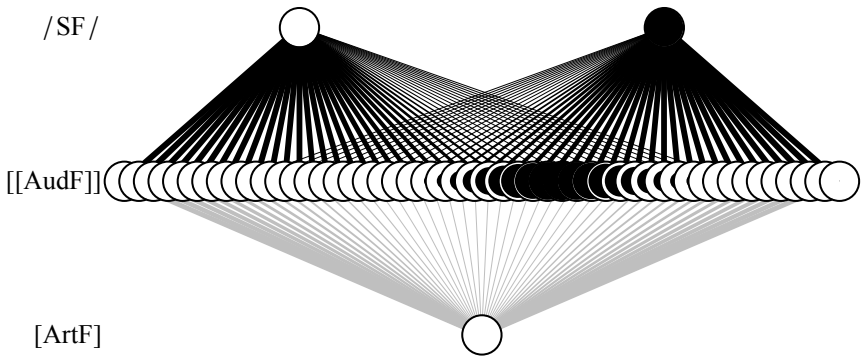

Fig. 20. The neural network after 50,000 learning steps.

The cue connections between the left half of the AudF layer and the left SF node are strongest, so the network has learned that low auditory values are most likely to be intended as this category; high auditory values, on the other hand, are mapped to the right-hand category, as the language environment dictated.

6.4. *Production*

The network is bidirectional, so it uses the same connections in production. Additionally, now the ArtF node comes into play, constraining the activities at the AudF layer. The SF node of the category to be produced is switched on and clamped, as well as the ArtF node; the node of the other category is switched off (and clamped). Figure 21 shows the resulting activities on the AudF layer (negative activities are clipped at zero) in the production of the left category. As expected, the activities in Fig. 21 are highest in the left half of the AudF layer. Note that while the cue connections to the peripheral AudF nodes are strong, meaning that these auditory values constitute non-confusable tokens of the category, these nodes are still inactive due to the large articulatory effort that is associated with them.

Following Boersma and Hamann (2008) we can define the *prototype* of a category to be the auditory form that is realized if the category is switched on at SF and there is no activity at ArtF. The prototype is therefore the AudF node that has the strongest cue connection to the category node. When ArtF is switched on, the node with the highest activity in production isno longer the prototype: the speaker realizes a more central auditory value than the prototype, because (observationally) she prefers values that are at the same time both auditorily sufficiently distinctive and easy to pronounce. This replicates the observation that listeners choose more peripheral tokens as prototypical than they produce themselves (Johnson, Flemming and Wright 1993; explained with BiPhon-OT by Boersma 2006).

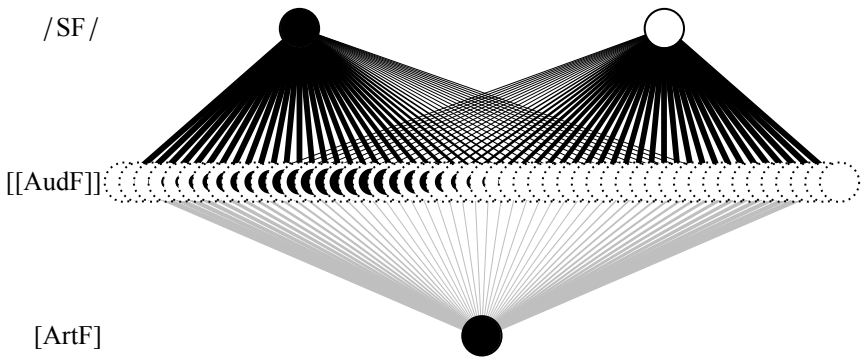

Fig. 21. Output activities for the left category (peaks in input distribution at 35 and 65%).

6.5. *Evolution over the generations*

The network is fairly insensitive to the input to the learning process. Whether the contrast between the categories is moderate or exaggerated, already in the first generation the activity patterns are very similar. Fig. 22 shows a network whose input distribution has peaks at 20% and 80% of the auditory continuum, rather than the 35% and 65% from Fig. 21, producing the left category. In spite of the peripheral location of the input, the activity pattern has shifted towards the center of the AudF level.

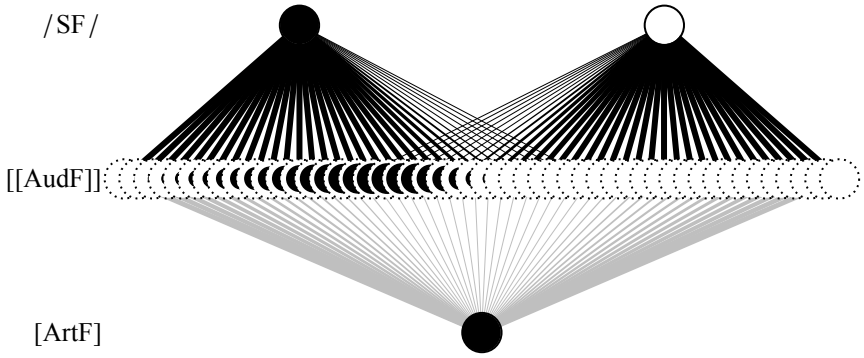

Fig. 22. Output activities for the left category (peaks in input distribution at 20 and 80%).

Fig. 23 shows the evolution of a standard input distribution (peaks at 35% and 65%, Figs. 20–21), which turns out to be very stable over the generations. The black lines connect the average produced nodes of a category, the gray lines connect the average produced node ± 1 standard deviation.

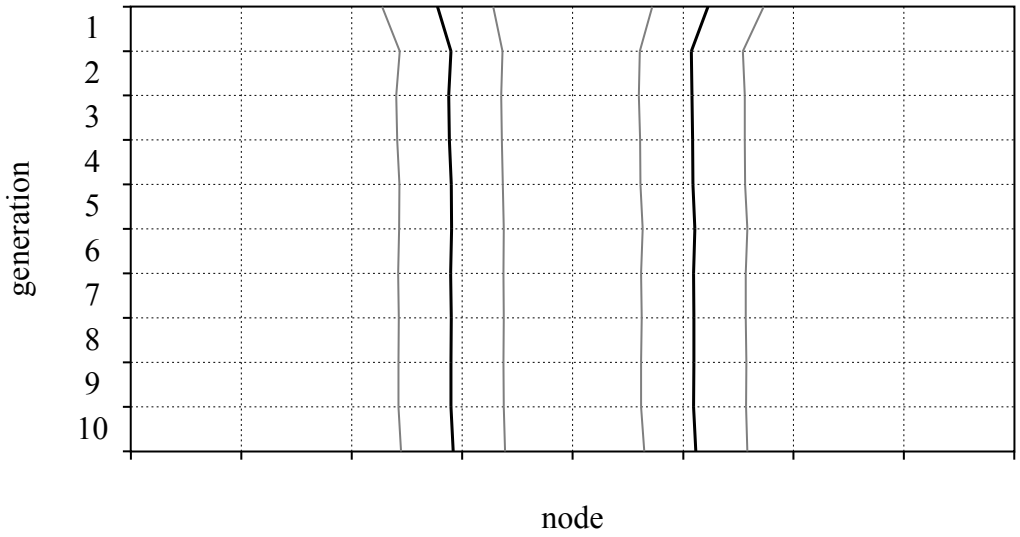

Fig. 23. Evolution of a two-category inventory with a standard input distribution.

Our simulations predict that an exaggerated system is diachronically unstable. Fig. 24 shows the evolution of such a distribution (peaks at 20% and 80%, Fig. 22). The first generation resolves the excessive contrast in the system; in fact, this generation even moves its production to a slightly more central region than the second generation, because the confusability in that region is lowest (see Fig. 22: cue connections to both categories emerge from the peripheral regions of the AudF layer). However, after the second generation, the system has evolved into the equilibrium seen in Fig. 23.

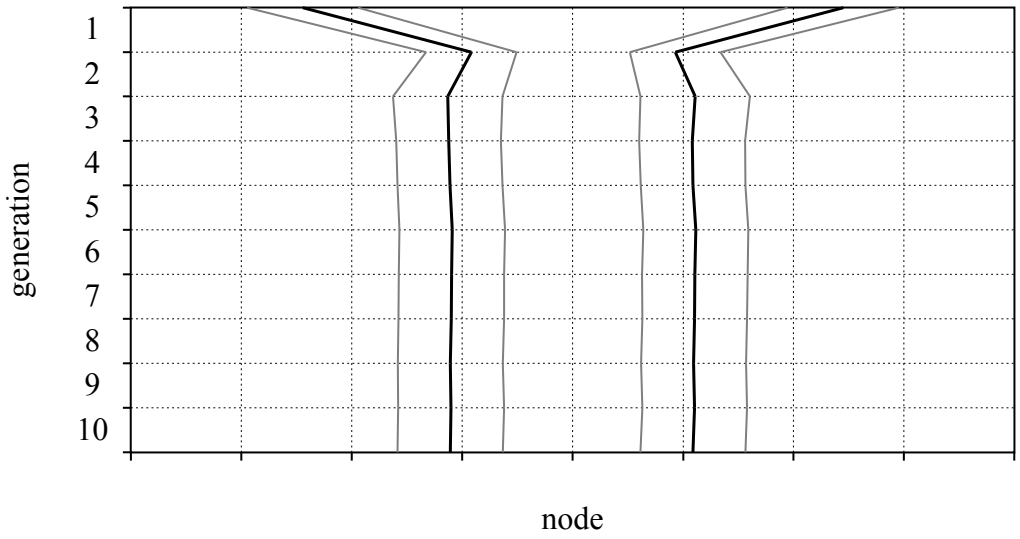

Fig. 24. Evolution of an exaggerated system (outstar learning).

Figure 25 shows the evolution of a skewed and confusing initial distribution, with peaks at 50% and 65%. Unlike the previous distributions, this inventory is not symmetric, and its categories lie quite close together. The first generation immediately enhances the contrast within the inventory by shifting the left category considerably towards the left periphery; only when this has happened does the right category drift downwards slightly. Because of the

overlap between the categories in the initial distribution, the standard deviations increase in the production of the first generation; they decrease again in subsequent generations, when the confusing contrast has been resolved. Within five generations, the system has reached the familiar symmetric and stable state.

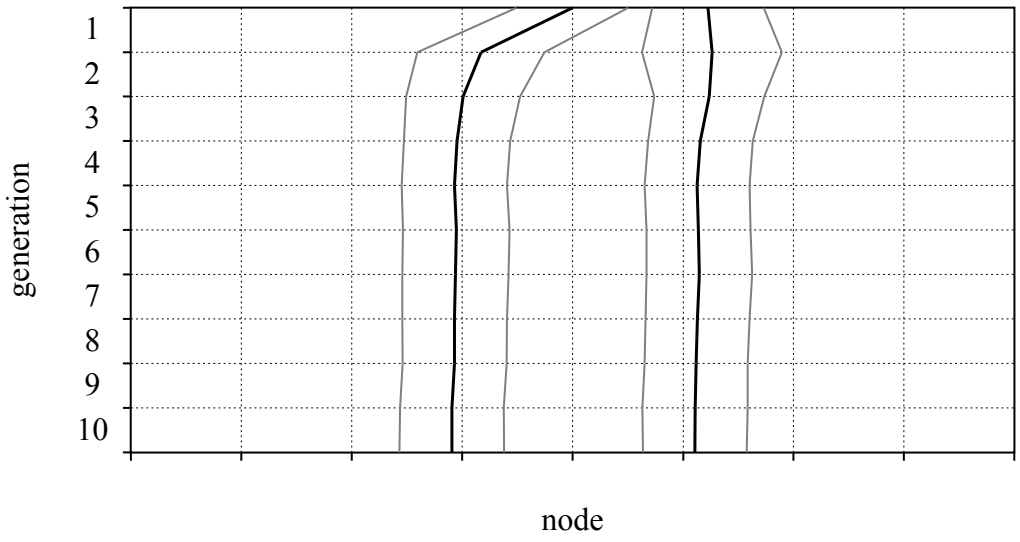

Fig. 25. Evolution of a skewed and confusing system (outstar learning).

6.6. *Influence of the learning rule*

When we use the inoutstar weight update rule instead of the outstar rule, only the cue connections from frequently activated AudF nodes increase in strength, irrespective of the predictability of the input-output relation in less frequently activated nodes. Hence, the connections are strongest at the peaks in the input. After 50,000 learning steps of a standard distribution (35%, 65%), the network thus comes to look differently from Fig. 20, namely as in Fig. 26.

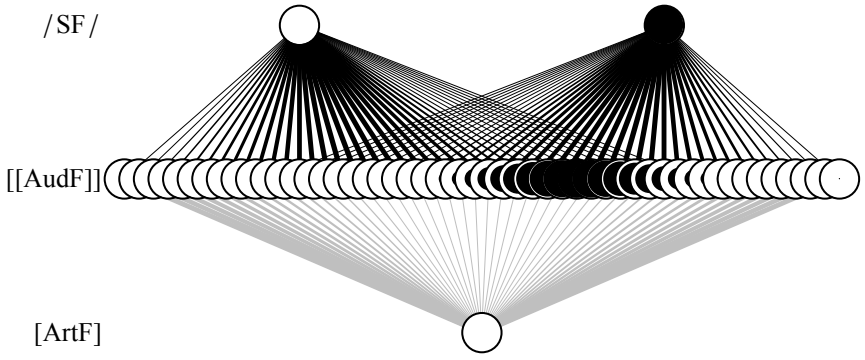

Fig. 26. A neural network after 50,000 learning steps (standard distribution, inoutstar learning).

The difference between outstar and inoutstar learning become most apparent in the evolution of exaggerated and confusing contrasts, which are resolved much slower with inoutstar learning. Whereas with outstar learning of an exaggerated input distribution, two generations sufficed to reach an equilibrium, inoutstar learning requires at least ten, as seen in Fig. 27.

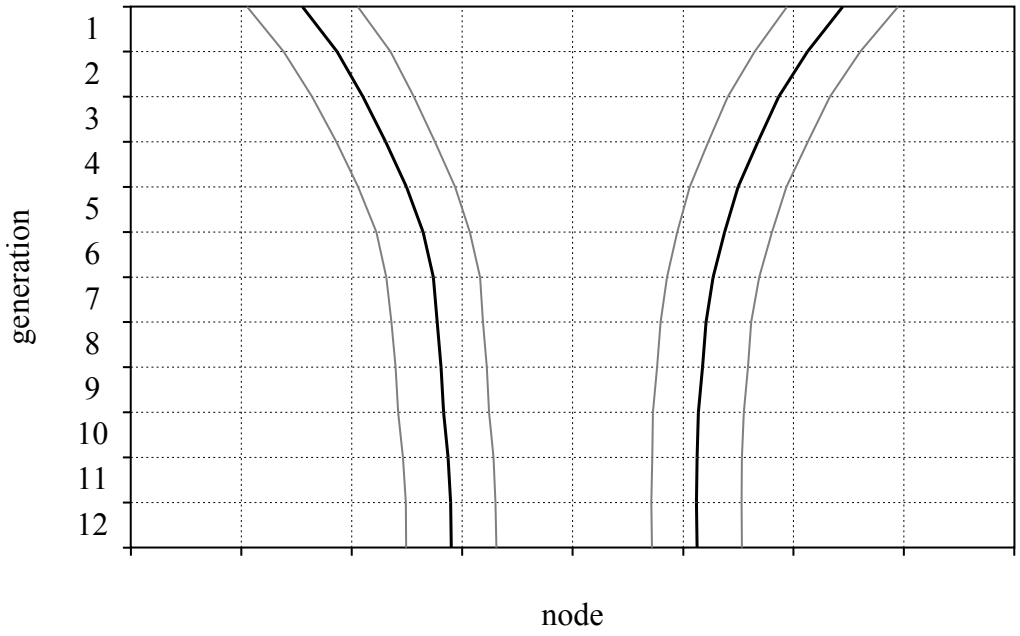

Fig. 27. Evolution of an exaggerated system (inoutstar learning).

The inoutstar rule also does not predict the quick phonetic enhancement of the contrast in a confusing inventory that the outstar rule predicted. As seen in Fig. 28, the categories drift apart slowly, and the between-category distance only reaches its maximum in the fourth generation. We witness the same increased standard deviation observed with outstar learning, but this does not decrease until the fourth generation. An optimally dispersed state has not yet been reached by generation 20, whereas five generations were enough with outstar learning. Nevertheless, inoutstar learning predicts optimally dispersed systems as well. It is an open question how fast real inventories get dispersed.

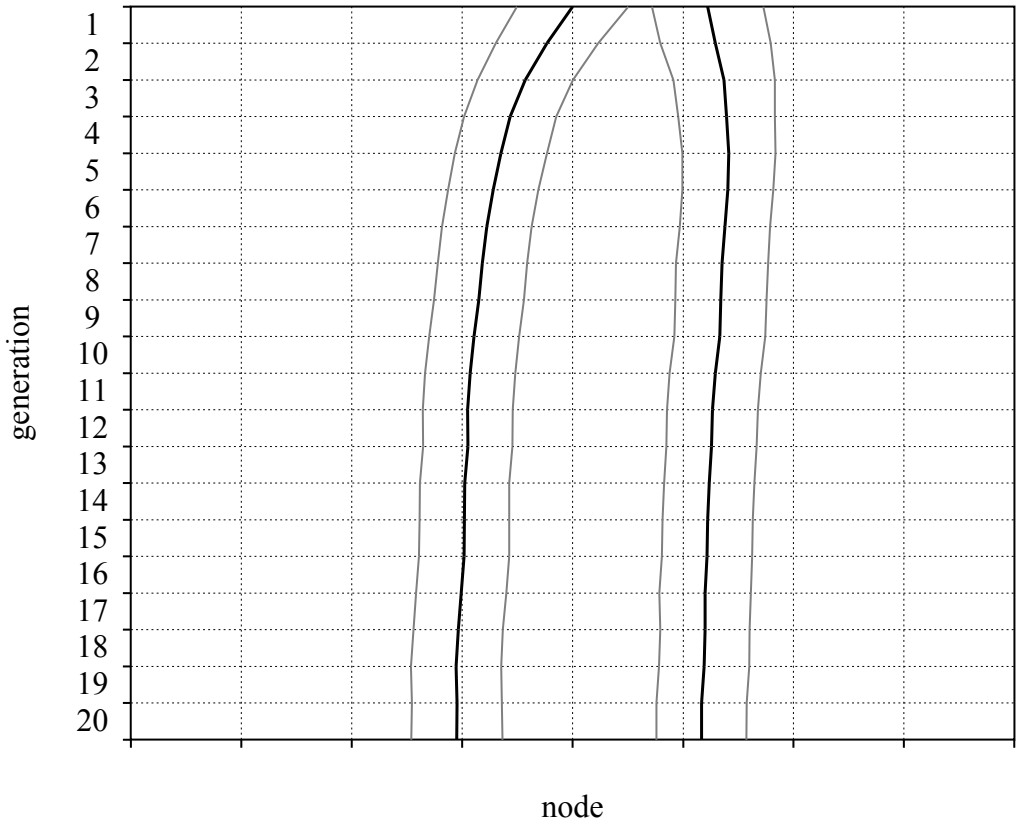

Fig. 28. Evolution of a skewed and confusing system (inoutstar learning).

In summary, our simulations show 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 using the same connections, any input distribution will evolve into a stable system within a small number of generations. It is thus crucial that the neural network is symmetric. The fastest results are obtained with the outstar weight update rule, but the inoutstar rule yields similar outcomes in the end.

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

7. Discussion

One and the same network, with a single learning rule, namely "inoutstar" learning, has turned out to be able to handle but category creation and auditory dispersion. While the instar rule is possible for category creation (as Gunether and Gjaja 1996 have shown), and the outstar rule is possible for the emergence of auditory dispersion (as shown here in §6), only the inoutstar rule, which is a combination of the instar and outstar rules, works for both.

The model achieves this success without having to represent or compute auditory distance. The interactivity of the processes is maintained because activity spreading in the neural network is interactive, i.e. simultaneously top-down and bottom-up, as in McClelland and Elman's (1986) TRACE model.

The model cannot really represent more than one segment yet: no phonological structure beyond single categories can be represented yet in the distributed versions of the network. This points at 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 have never been modeled without a single framework before. The BiPhon-NN model is also biologically more plausible than an OT model. One of the main missing areas involves strictly phonological phenomena, which would require the model to represent at SF sequential or hierarchical structures.

[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 016.094.610 to Boersma, and NWO grant xx.xx to Benders.

References

Apoussidou, Diana. 2007. The learnability of metrical phonology. Ph.D. dissertation, University of Amsterdam.

Blevins, Juliette. 2004. *Evolutionary Phonology: The Emergence of Sound Patterns.* Cambridge: Cambridge University Press.

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

Boersma, Paul. 1998. Functional Phonology: Formalizing the interactions between articulatory and perceptual drives. Ph.D. dissertation, University of Amsterdam.

Boersma, Paul. 2000. The OCP in the perception grammar. Ms. [Rutgers Optimality Archive **435**, http://roa.rutgers.edu]

Boersma, Paul. 2006. Prototypicality judgments as inverted perception. In Gisbert Fanselow, Caroline Féry, Ralf Vogel, & Matthias Schlesewsky (eds.), *Gradience in Grammar: Generative Perspectives*, 167–184. Oxford: Oxford University Press.

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

Boersma, Paul. 2009. Cue constraints and their interactions in phonological perception and production. In Paul Boersma & Silke Hamann (eds.), *Phonology in Perception*, 55–110. Berlin: Mouton De Gruyter.

Boersma, Paul. 2011. A programme for bidirectional phonology and phonetics and their acquisition and evolution. In Anton Benz & Jason Mattausch (eds.), *Bidirectional Optimality Theory*, 33–72. Amsterdam: John Benjamins.

Boersma, Paul. 2012. Modelling phonological category learning. In Abigail C. Cohn, Cécile Fougeron, & Marie K. Huffman (eds.). *The Oxford Handbook of Laboratory Phonology*, 207–218. New York: Oxford University Press.

Boersma, Paul, Paola Escudero, & Rachel Hayes. 2003. Learning abstract phonological from auditory phonetic categories: An integrated model for the acquisition of language-specific sound categories. *Proc. 15th ICPhS Barcelona*, 1013–1016.

Boersma, Paul, & Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32. 45–86.

Boersma, Paul, & Silke Hamann. 2008. The evolution of auditory dispersion in bidirectional constraint grammars. *Phonology* 25. 217–270.

Boersma, Paul, & Silke Hamann. 2009. Loanword adaptation as first-language phonological perception. In Andrea Calabrese & W. Leo Wetzels (eds.), *Loanword Phonology*, 11–58. Amsterdam: John Benjamins.

de Groot, Willem. 1931. Phonologie und Phonetik als Funktionswissenschaften. *Travaux du Cercle Linguistique de Prague* 4. 146–147.

Escudero, Paola, & Paul Boersma. 2004. Bridging the gap between L2 speech perception research and phonological theory. *Studies in Second Language Acquisition* 26. 551–585.

Flemming, Edward. 2002. Auditory Representations in Phonology. New York & London: Routledge.

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

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

Grossberg, Stephen. 1987. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science* 11. 23–63.

Guenther, Frank H., & Marin N. Gjaja. 1996. The perceptual magnet effect as an emergent property of neural map formation. *J. Acoust. Soc. Am.* 100. 1111–1121.

Hebb, Donald O. 1949. *The Organization of Behavior*. New York: Wiley.

Hopfield, John J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79. 2554–2558.

Jackendoff, Ray. 1997. *The Architecture of the Language Faculty*. Cambridge, Mass.: MIT Press.

Jackendoff, Ray. 2007. A parallel architecture perspective on language processing. *Brain Research* 1146. 2–22.

Johnson, Keith, Edward Flemming, & Richard Wright. 1993. The hyperspace effect: Phonetic targets are hyperarticulated. *Language* 69. 505–528.

Kirchner, Robert. 2001. *An Effort Based Approach to Consonant Lenition.* New York & London: Routledge.

Kuhl, Patricia K. 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.

Levelt, Willem, Ardi Roelofs & Antje Meyer (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences* 22. 1–75.

Liberman, Alvin M., Katherine Safford Harris, Howard S. Hoffman, & Belver C. Griffith. 1957. The discrimination of speech sounds within and across phoneme boundaries. *Journal of Experimental Psychology* 54. 358–368.

Lindblom, Björn. 1986. Phonetic universals in vowel systems. In John Ohala & Jeri Jaeger (eds.), *Experimental phonology*, 13–44. Orlando: Academic Press.

Martinet, André. 1960. *Éléments de linguistique générale*. Paris: Armand Colin.

McClelland, James L., & Jeffrey L. Elman. 1986. The TRACE model of speech perception. *Cognitive Psychology* 18. 1–86.

Ohala, John. 1981. The listener as a source of sound change. In C.S. Masek, R.A. Hendrick & M.F. Miller (eds.), *Papers from the parasession on language and behavior.* Chicago: Chicago Linguistic Society. 17. 178–203.

Ohala, John J. 1993. Sound change as nature's speech perception experiment. *Speech Communication* 13. 155–161.

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

Passy, Paul. 1890. *Etude sur les changements phonétiques et leur caractères généraux*. Paris: Firmin-Didot.

Pater, Joe. 2004. Bridging the gap between receptive and productive development with minimally violable constraints. In René Kager, Joe Pater, & Wim Zonneveld (eds.), *Constraints in Phonological Acquisition*, 219–244. Cambridge: Cambridge University Press.

Prince, Alan, & Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. Technical Report TR-2, Rutgers University Center for Cognitive Science. [published in 2004 by Blackwell, Malden Mass. & Oxford]

Rumelhart, David E., & David Zipser. 1985. Feature discovery by competitive learning. *Cognitive Science* 9. 75–112.

Seinhorst, Klaas. 2012. The evolution of auditory dispersion in symmetric neural nets. M.A. thesis, University of Amsterdam.

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

Steriade, Donca. 2001. Directional asymmetries in place assimilation. In Elizabeth Hume & Keith Johnson (eds.), *The Role of Speech Perception in Phonology*, 219-250. San Diego: Academic Press.

ten Bosch, Louis. 1990. On the structure of vowel systems: Aspects of an extended vowel model using effort and contrast. Ph.D. dissertation, University of Amsterdam.

Tesar, Bruce. 1997. An iterative strategy for learning metrical stress in Optimality Theory. In Elizabeth Hughes, Mary Hughes, & Annabel Greenhill (eds.), *Proceedings of the 21st Annual Boston University Conference on Language Development*, 615–626. Somerville, MA: Cascadilla.

Tesar, Bruce, & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29. 229–268.

Tesar, Bruce, & Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

von der Gabelentz, Georg. 1901. *Die Sprachwissenschaft: ihre Aufgaben, Methoden und bisherigen Ergebnisse*. Leipzig: Tauchnitz.

Wedel, Andrew. 2004. Self-organization and categorical behavior in phonology. Ph.D. dissertation, University of California at Santa Cruz.

Wedel, Andrew. 2006. Exemplar models, evolution and language change. *The Linguistic Review* 23. 247–274.

Wedel, Andrew. 2007. Feedback and regularity in the lexicon. *Phonology* 24. 147–185.