Generative linguistics and neural networks at 60: foundation, friction, and fusion

Abstract. The birthdate of both generative linguistics and neural networks can be taken as 1957, the year of the publication of seminal work by both Noam Chomsky and Frank Rosenblatt. This paper traces the development of these two approaches to cognitive science, from their largely autonomous early development in their first thirty years, through their collision in the 1980s around the past tense debate (Rumelhart and McClelland 1986, Pinker and Prince 1988), and their integration in much subsequent work up to the present, 2017. Although these traditions are often presented as in opposition to one another, such a presentation assumes polar versions of each approach, and ignores the ever-growing body of results that have been achieved through integration.

1. Introduction
At the beginning of 1957, two men nearing their 29th birthdays published work that laid the foundation for two radically different approaches to cognitive science. One of these men, Noam Chomsky, continues to contribute sixty years later to the field that he founded, generative linguistics. The book he published in 1957, Syntactic Structures, has been ranked as the most influential work in cognitive science from the 20th century.1 The other one, Frank Rosenblatt, had by the late 1960s largely moved on from his research on perceptrons – now called neural networks – and died tragically young in 1971. On the same list of 100 influential works that ranked Syntactic Structures #1, there is nothing by Rosenblatt, though Perceptrons, the 1969 book by Minsky and Papert, is listed at #74. Rosenblatt’s 1957 tech report, “The perceptron: a perceiving and recognizing automaton”, is very short and fairly programmatic, but the line of research that it began, much of it presented in his 1962 book, Principles of Neurodynamics, launched modern neural network modeling, including the multi-layer perceptrons used in deep learning (LeCun, Bengio & Hinton 2015).

Chomsky and Rosenblatt’s approaches to cognitive science were radically different because they started at opposite ends of the problem of developing a computational theory of the human mind. Chomsky took a “high level” cognitive phenomenon – language, and in particular, syntax – and aimed to show that some reasonably powerful computational machinery was not up to the task of representing it, before going on to propose a more powerful theory that could. Rosenblatt took some very simple computational machinery – mathematical analogues of neural activation and synaptic connections – and aimed to show that it could represent “low level” cognitive processes involved in object perception and recognition, and that these representations could be learned algorithmically.

---

1 The list “The Top 100 most influential works in cognitive science from the 20th century” was compiled by the Center for Cognitive Sciences, University of Minnesota, in 2000 (Sanz 2008). I realize that picking any date, and any person, as the beginning of a tradition is somewhat arbitrary. Many of the ideas in Chomsky (1957) and Rosenblatt (1957) can be traced back much earlier (Pullum 2011; Schmidhuber 2015 respectively). But Chomsky and Rosenblatt’s work was clearly particularly prominent and influential at the time, and did sow the seeds for much subsequent research in generative linguistics and neural networks.
Although there is no evidence that Chomsky and Rosenblatt ever met, or that either one’s research had an impact on the other, both of them interacted with members of the intellectual community working on what was dubbed Artificial Intelligence (AI) in 1956 by John McCarthy and Marvin Minsky. Chomsky’s arguments about the representational complexity of language were made in the context of the models being explored by that community, in particular Finite State Automata, and related probabilistic Markov chains. George Miller gives the date of a presentation of that work at an AI conference, September 11, 1956, as the birthdate of cognitive science (Miller 2003:142). Rosenblatt presented an early version of his perceptron research at MIT to an AI group in the fall of 1958 (Jack Cowan in Anderson & Rosenfeld 2000:99–100). Amongst the members of that audience was Minsky, Rosenblatt’s high school colleague, and the future first author of the critical appraisal of perceptrons mentioned above. Section 2 discusses how Chomsky and Rosenblatt’s proposals each diverged from “mainstream AI”.

Generative linguistics and neural network modeling developed in apparently complete isolation from one another until they collided thirty years later, when Rumelhart and McClelland (1986) developed a perceptron-based, or connectionist, model of past tense formation in English, which was fiercely criticized from a linguistic perspective by Pinker and Prince (1988) and others. The broader debate between proponents of “algebraic” approaches to cognition, like generative linguistics, and “analogical” models, like connectionism, defined much of the landscape of cognitive science as it developed in the late 1980s and early 1990s. The second section of the paper discusses some of that debate, and argues that it produced important lessons for future research in both traditions, rather than ending in victory for one or the other side. I also survey some of the characteristics of each of the paradigms, and point out that although they can usefully differentiate them, their use in contrastive definitions can also lead to false dichotomies.

The final section of the paper surveys some of the subsequent research over the last thirty years that has integrated aspects of neural network modeling research and generative linguistics. One instance of such integration is Optimality Theory (Prince and Smolensky 1993/2004), whose founders were on opposite sides of the debates of the 1980s. I also discuss the recent resurgence of interest in neural networks in AI, and the emergence of renewed study in cognitive science and AI of their ability to represent linguistic generalizations, with or without explicitly coded linguistic representations.

2. Foundation

2.1 Neural networks

Rosenblatt’s (1957) perceptron uses analogues of neural structure developed in earlier work, in particular by McCulloch and Pitts (1943) and Hebb (1949). The activity of a neuron – also called a node or a unit – is represented in numerical terms, often as 1 or 0, on or off. This activity is passed along a synaptic connection to other neurons. The connections are weighted: each one has a real valued number that is multiplied by the signal it receives from an input neuron. A given neuron becomes active when the sum of incoming weighted signals exceeds a designated threshold (given a step-activation function, rather than for instance a sigmoidal activation function).

The following table represents a small perceptron being used to classify a small set of objects. There are two features “+Black” and “+Star”, each of which defines an Input node. In Rosenblatt (1957 et seq.) these are the A units (for Associative), which
would have themselves been activated through connections to S units (for Sensory). The weights on their connection to a single neuron (an Output, or Response unit) are shown beneath their names in the top row: 0.75 and 0.34 respectively. In each row we see the Input node activations for each of the four objects. The weighted sum shows the total signal received by the neuron. With these weights, only the black objects activate the neuron, given an activation threshold of 0.5.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Input} & \text{+Black} & \text{+Star} & \text{Weighted Sum} & \text{Activation (> 0.5)} \\
\hline
\star & 1 & 1 & 1.09 & 1 \\
\bullet & 0 & 1 & 0.34 & 0 \\
\diamond & 1 & 0 & 0.75 & 1 \\
\hline
\end{array}
\]

(1) A perceptron classifying the set of black objects

Different positive and negative weights will lead to different sets of objects activating the neuron. Famously, the sets characterized by an “exclusive or” logical relation (XOR) cannot be picked out by this type of network (see Minsky & Papert 1988 for a proof). For instance, the objects in the middle two rows, the white star and the black triangle, cannot be the only ones to activate the Output neuron. These two objects form an XOR set: “+Black, or +Star, but not both”. One way to get this classification would be to include an Input node that is activated by the conjunction of the features [+Black] and [+Star]. The weight on the connection from that node could then be given a negative value sufficiently high that the black star’s activation falls beneath the 0.5 threshold, even while the other black object and the other star’s activation are above it. The classificatory power of this simple type of perceptron is dependent on content of the Input nodes: with arbitrarily complex combinations of features, it can perform arbitrarily complex classifications.

In current parlance, this type of perceptron is called a single-layer neural network: there is a single set of adjustable weights connecting the Input layer of nodes to the Output layer (which was in our case just a single node).\(^2\) The general form of a single layer network is illustrated in fig. 1 (from Olazaran 1993; see Elman et al. 1996; and Marcus 2001 for introductions to neural nets for cognitive science).

\(^2\) These are sometimes called two-layer networks (e.g. Pinker and Prince 1988, Marcus 2001), since there are two layers of nodes (Input and Output). This has the downside of making the term “multilayer” opaque.
Besides increasing the size of the Input layer, another way to capture the XOR class is to introduce an additional “hidden” layer of neurons intermediate between the Input and Output, as shown in the following pair of tables. The first table shows the mapping from the Input to the hidden layer, which consists of a pair of neurons. The weights to each of them are arrayed vertically, as are the resultant weighted sums and activations. For example, the first neuron has feature weights of 0.26 and 0.27, which results in a weighted sum of 0.53 for the black star, and activation only for that object (this implements logical AND).

(2) Multi-layer perceptron part 1: Input to hidden layer

<table>
<thead>
<tr>
<th>Input</th>
<th>+Black</th>
<th>+Star</th>
<th>Weighted Sum</th>
<th>Activation (&gt; 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠</td>
<td>1</td>
<td>1</td>
<td>0.53</td>
<td>1</td>
</tr>
<tr>
<td>♣</td>
<td>0</td>
<td>1</td>
<td>0.27</td>
<td>0</td>
</tr>
<tr>
<td>♦</td>
<td>1</td>
<td>0</td>
<td>0.26</td>
<td>0</td>
</tr>
<tr>
<td>♦</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The activation of the neurons in the hidden layer is the input for the next table. Since the first node is only active for the black star, it is acting like the conjunctive node discussed above. Its negative weight is sufficient to counteract the positive weight of the second node, which is active for objects that are either +Black or +Star (logical OR).
(3) **Multi-layer perceptron part 2: Hidden layer to Output**

<table>
<thead>
<tr>
<th>Input</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Weighted Sum</th>
<th>Activation (&gt;0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>★</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>⊖</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>♦</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>♢</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

This example shows that for a given set of input nodes, a multi-layer perceptron, that is, a network with a hidden layer, can have greater representational power than a single-layer one. Single-layer networks are limited to linearly separable patterns, while as the XOR example shows, multi-layer perceptrons can represent non-linearly separable patterns (separable in the two-dimensional space defined by the two feature values). For readers familiar with regression models, a useful analogy may be that the crucial node in the hidden layer is acting as an interaction term.

The greater representational capacity of a multi-layer perceptron comes at a price: it is more difficult to train. Rosenblatt (1957; 1958) developed a learning procedure for single-layer perceptrons that is guaranteed to find a set of weights that yields the desired pattern of activation, if such a set of weights exists (Block 1962; Novikoff 1962; Minsky & Papert 1988). As discussed in section 3.1, this procedure does not work with a hidden layer; the development of relatively successful and efficient learning algorithms for multi-layer perceptrons was a primary factor in the sharp increases in research using neural networks in the 1980s and the 2010s.

In a book originally published in 1969, but circulated before that, Minsky and Papert (1988) provide an in-depth study of the representational capacities of single-layer perceptrons, with a primary aim of showing the limits on what this type of perceptron can represent. The appearance of Minsky and Papert’s book coincided with a shift of focus in the mid-to-late sixties from neural networks to other branches of the new discipline of AI, in particular, ones that manipulate symbolic representations with algebraic operations (see Nilsson 2010 for an excellent history of AI by a participant in this shift). It is hard to know to what degree it was the force of Minsky and Papert’s observations and arguments that led to this sea change, or whether it was that neural net research was largely running out of steam given the theoretical and technical barriers that existed at the time; see Olazaran (1993; 1996) for a particularly useful discussion. It is worth emphasizing that Rosenblatt and his group also worked with multi-layer networks, since this is often overlooked: see Pater (2017) for examples of miscitation, and some speculation about the roots of this confusion, and see Rosenblatt (1962), Block et al. (1962), and Block (1970) for discussion of multi-layer perceptrons, as well as Nagy (1991) for a useful concise overview of research by Rosenblatt’s group.

Rosenblatt was trained as psychologist rather than as an engineer, and was committed to using perceptrons to model human psychological properties as realized in the brain, as the following citation from his 1962 book emphasizes (p. vi):

> A perceptron is first and foremost a brain model, not an invention for pattern recognition. As a brain model, its utility is in enabling us to determine the physical conditions for the emergence of various psychological properties. It is by
no means a “complete” model, and we are fully aware of the simplifications that have been made from biological systems; but it is, at least, an analyzable model.

Despite Rosenblatt’s characterization of perceptrons as brain models underlying cognition, research on neural networks did not have much, if any, impact on the emerging field of cognitive science. A 1978 report on cognitive science commissioned by the Sloan foundation to survey the field lists no such research in its bibliography (Keyser, Miller & Walker 1978). Shortly after that report was written, there would be a quite dramatic upsurge in neural network modeling, and at that point, it became a core part of cognitive science. These developments are covered in section 3.

2.2. Foundations: generative linguistics

Chomsky (1957:13) sets up the analysis of a language as a classification problem:

The fundamental aim in the linguistic analysis of a language \( L \) is to separate the grammatical sequences which are the sentences of \( L \) from the ungrammatical sequences which are not sentences of \( L \) and to study the structure of the grammatical sequences.

Distinguishing a well-formed sentence from an ill-formed one differs from the object classification examples discussed in the last section in a number of ways. In particular, order in time matters for a sentence, whereas in the examples discussed above, there was no temporal or spatial relationship between the features.\(^3\) A simple example of the importance of order in syntax is that *The boy sleeps* is a well-formed English sentence, but *Sleeps boy the* is not. There are also dependencies between items that occur at different points in time, as in subject-verb agreement: *e.g. The boy sleeps and The boys sleep vs. *The boy sleep and *The boys sleeps.*

One way of encoding the difference between these grammatical and ungrammatical sentences is in terms of allowable transitions between words: *the* followed by *boy* is permitted, but not the reverse, and *boy* can be followed by *sleeps* but not *sleep.* Chomsky (1957: ch. 3) shows that “[a] familiar communication theoretic model for language” (p. 18) can encode these sorts of restrictions. This model, a finite state machine, specifies a set of states of machine, along with allowable transitions between them. For example, we can specify that if a machine is in the “boy” state, it may move into the “thinks” state, but not the “think” state.

As the sentence below makes clear – as does the name of the field that he was laying the groundwork for – Chomsky (1957: 13) proposes to classify sentences as grammatical and ungrammatical in terms of whether or not they are generated by a grammar:

---

\(^3\) Goldberg (2012) makes use of this property of simple perceptrons to show that baboons’ knowledge of legal vs. illegal orthographically presented English words can be modeled without reference to word position or linear order, contrary to a claim in the original study (Grainger et al. 2012). See Rosenblatt (1967) for an interesting proposal about how to encode memory through time in a more complex perceptron, one that has some similarities to the recurrent neural networks and convolutional networks used in contemporary deep learning (see section 4.2).
The grammar of $L$ will thus be a device that generates all of the grammatical sequences of $L$ and none of the ungrammatical ones.

As a generative device, a finite state grammar (FSG) can generate an infinite number of sentences using finite resources, a basic criterion for adequacy that Chomsky sets up for a theory of language. It can do this because it allows for loops. Figure 1 provides an FSG for an example from Chomsky (1957), which generates “the man comes”, “the old men come”, “the old man comes”, “the old, old, man comes” and so on. The first node “the” is the start state, and arrows show permissible transitions, with “man” being the end state.

![Figure 1: Finite State Grammar](image_url)

Chomsky (1957) cites Shannon and Weaver’s (1949) presentation of information theory as the source of the finite state formalism. While an FSG can represent some ordering restrictions and dependencies amongst the words of a grammatical sentence, Chomsky (1957: 21-25) argues that it cannot represent the full complexity of English sentence structure. In particular, Chomsky point to the fact that English allows nested dependencies between particles like “if … then” and “either … or”:

(4) If John either ate or drank then he couldn’t sleep.
   *If John either ate then drank or he couldn’t sleep.

The grammatical sentence exhibits a mirror structure: if we notate the dependent particle in lower case, the correct structure is $ABba$, and the ungrammatical sentence is instead $ABab$. An FSG cannot generate unboundedly many of such mirrored structures, that is, it cannot represent $A^aB^b$ (see the recent discussion in Jäger and Rogers 2012, as well as the critical commentary on Chomsky 1957 in Pullum 2012). Interestingly, one of the challenges for perceptrons that Minsky and Papert (1988) discuss is the similar recognition of symmetry (pp. 117, 252-254). I should note that “mirror structure” is not the conventional term for the linguistic phenomenon, rather, it is usually discussed as

---

4 Cherry (1957) provides a useful overview of the state of the art in language modeling at the time, with introductions to structural linguistics and information theory; this book was required reading for the first class of MIT PhD students in Linguistics (Barbara Partee, personal communication 2016). In the context of the present joint history of generative grammar and neural networks, it is worth mentioning that Feldman’s (1992: 73) encyclopedia entry on finite state machines locates their origin in McCulloch and Pitts’ (1943) study of the logical power of mathematical analogues of neurons, and also that Kleene’s (1956) introduction of the connection between finite state machines and regular expressions was in the context of “clarifying the results of McCulloch and Pitts” (Block 1970: 513-514).
“center embedding”, and a more typical example would be center embedded relative clauses, such “The boy that hit the balls that flew over the fence is cheering.”

The discussion of the limitations of FSGs in Chomsky (1957: sec. 3) comes after a discussion of the inadequacy of “statistical approximation to English” as a proxy for the grammatical / ungrammatical distinction (sec. 2). This takes as its starting point the famous pair of sentences in (5).

(5) Colorless green ideas sleep furiously.
    Furiously ideas sleep green colorless.

Chomsky (1957) claims that since the two sentences, and their subparts, would equally be zero frequency in a corpus, they would be “equally ‘remote’ from English” “in any statistical model of English”. Pereira (2000) has shown that this is incorrect: these sentences are in fact distinguished by a bigram model over word categories (note for example that the second sentence starts with a less frequent adverb-noun sequence, compared with the adjective-noun sequence in the first). Nonetheless, Chomsky (1957: 24) is undoubtedly right that it would be a “dead end” to try to reduce sentence wellformedness to n-gram probabilities over sequences, or to the more complex distributions over sequences that can be represented by a probabilistic Markov chain instantiation of an FSG, regardless of whether those sequences are of words or categories.

So what characteristics must an adequate grammatical model for English have? One obvious characteristic that goes beyond mere sequential restrictions is the ability to represent the hierarchical phrase structure of a sentence. Chomsky (1957: 30) shows that a phrase structure grammar is able to generate the nested dependencies discussed above, and is thus more powerful than an FSG. Chomsky’s proposal goes further than phrase structure in also making use of transformations, which take as input a string with phrase structure and produce as output another string with a new constituent structure (p. 44). Chomsky argues that by deriving passive sentences, negation, and questions through transformations on a base “kernel” phrase structure, considerable simplifications in the form of the grammar can be obtained.

The relatively deep derivations entailed by a transformational grammar were seen as a liability by some, and alternative frameworks emerged that eschewed them (GPSG, LFG, HDPSG). Deep derivations have generally been retained in the frameworks that Chomsky himself has subsequently developed, including current minimalism, although the objects being manipulated by the derivations have changed. For a useful recent comparison of minimalism with two of its non-derivational competitors, LFG and HPSG, see Bond et al. (2016).

The postulation of these two kinds of abstract structure – hierarchical constituents, and underlying forms that are derivationally transformed into the surface structure – are also characteristic of generative analyses of aspects of language other than syntax. Even though phonological restrictions seems to be representable at the segmental string level by FSGs (Heinz & Idsardi 2011; Heinz & Idsardi 2013), hierarchical representations are standardly viewed as necessary for an adequate characterization of the phonologies of the world’s languages (Selkirk 1981; Yu 2017). A well-known example of derivational transformation in phonology is the postulation of an abstract underlying /ai/ in words like title and writer in Canadian English, in which they are pronounced with a ‘raised’
diphthong [ai] that contrasts with surface [ai] in words like bridle and rider. The two surface phones can be generated from a single underlying phoneme by assuming that as in the spelling, title and writer have an underlying /t/, which conditions raising before it becomes the surface flap (Harris 1951; Chomsky 1964). As in syntax, the derivational depth of many analyses in generative phonology has been controversial (see Anderson 1985 for an overview of this controversy).

In phonology at least, one reason that deep derivations are controversial is that they are suspected to pose difficulties for learning (see Dresher 1981 for a critique of this view). Here we can draw a potentially useful connection to the hidden layers of neural nets, discussed in the last section. The perceptron analysis of XOR pattern classification made use of a hidden layer node that abstractly specified the conjunction over features: it was not present in the “surface” Input nodes, which were activated by single features. Like the derivations and hierarchical structure of a generative linguistic analysis, the weights that lead to the conjunctive activation of the hidden layer have to be inferred by a learner. Tesar and Smolensky (2000) call linguistic structure that is not apparent to the learner “hidden”, and discuss the learning challenges that it poses (see further 4.1 below). Drawing a connection between hidden linguistic structure and hidden layers of neural nets is potentially useful in two ways, on which I’ll expand below. The first is that techniques for learning with hidden layers are potentially useful in learning with explicitly encoded hidden linguistic structure. The second is that hidden layers may be used to learn representations that take the place of explicitly encoded hidden structures. As a historical note, we can also draw a parallel between skepticism about hidden linguistic structure based on learning concerns, and Minsky and Papert’s (1988) skepticism about the tractability of learning with hidden layers.

It is this postulation of language-specific abstract structure that differentiates work in generative grammar from what one might call mainstream AI, which tends to postulate relatively concrete domain-general representations. As we will see in the discussion of connectionism in the sections that follow, generative linguistics is similar to mainstream AI in its use of algebraic operations over symbolic representations.

It is perhaps worth reminding ourselves that even the relatively concrete features of a surface linguistic representation, as well as the features used in the neural net discussed in the last section, are themselves abstractions from raw perceptual experience. At a minimum, a learner must acquire a mapping from a more basic level of representation to these features (insofar as it is culturally specific), and at least some of these features probably need to be induced. Given the usefulness of neural networks in vision and speech recognition applications in AI, they are especially likely to be useful in addressing this part of the learning problem in cognitive science. I will have no more to say about it here though, as my focus is on the “higher level” learning problems that are the focus of generative linguistics.

3. Friction
3.1 Innatism and emergentism
One of the main themes of this section is a caution against the false dichotomies that can be created by contrastively labeling neural network and generative linguistic research, so it is with some trepidation that I start this subsection with a heading that contains two of those very labels. However, to finish setting the stage for the encounter between the
generative and neural network traditions that occurred in the 1980s, it is important to emphasize how central learning had become in generative linguistics, and how dramatically the approaches to learning differed in the two traditions.

In Chomsky (1957), there is no real mention of learning, though his future emphasis on learnability considerations is foreshadowed in two ways. The first is in his discussion of the inadequacies of (probabilistic) sequential models of language, which can be trained by relatively simple learning algorithms. The second is in a section on the “Goals of Linguistic Theory”, in which he rejects the structuralist insistence on discovery procedures – algorithms for proceeding from a corpus to an analysis – in favor of a weaker requirement that there be an evaluation procedure, a method for choosing amongst hypothesized grammars. In Chomsky (1957), the evaluation procedure is for the linguist’s task of choosing amongst analyses and theories, but it would later be taken to be part of the acquisition process (Chomsky 1965, Chomsky and Halle 1968).

Shortly after the publication of Syntactic Structures, learning did of course become an explicit focus of Chomsky’s attention, in his 1959 review of Skinner’s (1957) *Verbal Behavior*, which contains the following passage (Chomsky 1959:sec. V):

It is often argued that experience, rather than innate capacity to handle information in certain specific ways, must be the factor of overwhelming dominance in determining the specific character of language acquisition, since a child speaks the language of the group in which he lives. But this is a superficial argument. As long as we are speculating, we may consider the possibility that the brain has evolved to the point where, given an input of observed Chinese sentences, it produces (by an induction of apparently fantastic complexity and suddenness) the rules of Chinese grammar, and given an input of observed English sentences, it produces (by, perhaps, exactly the same process of induction) the rules of English grammar; or that given an observed application of a term to certain instances, it automatically predicts the extension to a class of complexly related instances. If clearly recognized as such, this speculation is neither unreasonable nor fantastic; nor, for that matter, is it beyond the bounds of possible study. There is of course no known neural structure capable of performing this task in the specific ways that observation of the resulting behavior might lead us to postulate; but for that matter, the structures capable of accounting for even the simplest kinds of learning have similarly defied detection.

From this initial speculation about rapid induction, the generative program eventually became one of mapping out the hypothesis space that a learner was claimed to deductively navigate in acquiring a language – of characterizing Universal Grammar (UG). In principles and parameters theory (Chomsky 1981: 3-4), the hypothesis space is characterized by a set of universal principles, alongside language-specific parameters that are “fixed by experience”. The postulation of a relatively rich innate endowment is justified in learnability terms: “UG must be sufficiently constrained and restricted in the options it permits so as to account for the fact that each of these grammars develops on the basis of quite limited experience” (see Pullum and Scholz 2002 and the replies in the same volume for a relatively contemporary discussion of poverty of the stimulus, as well as the comments in Fereira 2000: 1243-1245 from an information theoretic perspective,
and the recent connectionist proposals in Fitz and Chang 2017, discussed further in 4.2 below).

In the just-cited passage, Chomsky argues for a restrictive theory of UG from learning considerations, but a restrictive UG-based theory of language typology has been taken as an independent goal in much subsequent generative research in both syntax and phonology. Because of this focus on restrictiveness, generative critiques of connectionist models would thus not only point out failures to acquire linguistic systems, but also their ability to learn patterns that fall outside those attested in human language (see 3.2 below).

Principles and parameters theory led to not only to a large body of UG-based typological research, but also considerable research on first and second language acquisition, as well as theories of learning addressing the question of how parameters are set (see e.g. papers in Roep & Williams 1987). The parameter-setting problem is non-trivial, and I will return to it in section 4.1 in the context of alternative models of linguistic typology.

At about the same time as principles and parameters theory emerged, neural network research was making large strides in the development of algorithms for the navigation of the representational space provided by multi-layer perceptrons. Rosenblatt’s (1957, 1958) learning algorithm for single-layer perceptrons is an error correction procedure: given an input, the network is used to predict an output. If the network’s output fails to match the correct output, yielding an error, the weights are changed slightly in the direction of generating the correct activation pattern. The activation pattern of a hidden layer is not given as part of the training data, so it is not straightforward to update its input weights. Rosenblatt could see that a solution would be to propagate the error signal back through the hidden layer (and even used the term backpropagation; Rosenblatt 1962: 297-298, cited in Olazaran 1993: 391) but neither he nor any of his contemporaries could not find an effective way of doing so. According to Terrance Sejnowski (interview cited in Olazaran 1993: 398), the crucial step in developing the algorithm known as backpropagation (Werbos 1982; Rumelhart, Hinton & Williams 1986; LeCun 1988; see Olazaran 1993; Schmidhuber 2015 on the history) was to replace the step activation function, which yields discrete activation levels, with a continuous sigmoidal function. This allowed for the calculation of a gradient, which determines the direction of the weight updates. Although back propagation does not have a convergence guarantee, it is (perhaps surprisingly) effective, and a variety of methods exist for increasing the likelihood that it will find a global, rather than a local, minimum of error.

Because of its focus on learned representations, neural network research is a largely emergentist tradition, and the connectionist linguistic literature often contrasts itself with Chomskyan innatism (Elman et al. 1996; Bates et al. 1998). It is important to emphasize, though, as the just-cited authors do, that just as innate parameters need a learning theory to respond to experience, a network needs a specified structure for it to represent the effects of learning.

3.2 The past tense debate

Even though it was the development of backpropagation for learning with hidden layers that launched the revitalization of neural network research in the 1980s, the model that became the focus of the debate between generativists and connectionists was a single
layer network, trained by a version of Rosenblatt’s (1957) perceptron update rule. The Input to this network is a phonological representation of uninflected form of an English verb, and its Output is the predicted phonological form of its past tense. Rumelhart and McClelland (1986:217) present this network as illustrating an alternative to the views that “the rules of language are stored in explicit form as propositions, and are used by language production, comprehension and judgment mechanisms” and that in learning, “[h]ypotheses are rejected and replaced as they provide inadequate for the utterances the learner hears”, using a learning mechanism that has “innate knowledge of the possible range of human languages”. That is, they set up their past tense simulation as illustrating an alternative to the generative approach to language and its learning, for which they cite Pinker (1984) as a state-of-the-art example.

Rumelhart and McClelland’s model of past tense formation represents an alternative to the generative view in the sense that there is neither an explicit rule that adds a past tense morpheme to an uninflected stem, nor are there explicit rules for determining the phonological shape of that morpheme (though see Elman 1991:218 for critical discussion of the implicit/explicit distinction). In fact, there is no morphology at all in the model besides the fact that the Input is itself the uninflected stem; there is no morphological decomposition in the Output past tense or any intermediate form. Both the addition of the phonologically appropriate form of “-ed” in regular past tense formation, as well as the various forms of irregular past tense such as vowel change (sing, sang), no change (hit, hit), and suppletion (go, went) are all handled by a single network of weighted connections between the atoms of the phonological representation of the uninflected form and those of the inflected one. Learning consists of weight adjustments in response to errors in the prediction of the past tense form. This is in fact a form of hypothesis testing in the sense that the current values of the weights represent the network’s current hypothesis, which is modified by a weight update, but it is different from most generative learning algorithms in that the hypotheses are over a continuous rather than a discrete space, and the changes in output are typically gradual, rather than abrupt (though see Elman et al. 1996:chap. 4 on nonlinearities in learning curves). Rumelhart and McClelland (1986) show that these gradual weight adjustments allow them to model the trajectory of the acquisition of the past tense, producing the U-shaped development that results from initial accurate production of irregulars, followed by over-regularization (goed, hitted), and then back to the correct target form.

One of the targets of Pinker and Prince’s (1988) critique of Rumelhart and McClelland (1986) is the nature of the phonological representations used in the model. Like the Input layer of the object classification networks in section 2 above, there is no temporal order in the Input nodes of the past tense model (see Elman 1990 for a discussion of the difficulties with a temporally ordered Input layer). To represent phonological contexts, a single node encodes features of both the preceding and following phone, as well as the central phone — a triphone representation called a Wickelfeature after Wickelgren (1969). This clearly not a general solution to the problem of encoding temporal order, and as Pinker and Prince (1988) discuss at length, it runs into number of problems. Amongst these is the fact that it can represent string reversals (mad → damn) as easily as it can represent an identity map, yet no language uses string reversals as a phonological process. Subsequent connectionist models of the past tense, as well as other types of morphophonology (see Alderete & Tupper 2018 for an overview),
tend to adopt one of two general approaches. One is to notate features for where appear in word, as in the templates of Plunkett and Marchman (1993). Another is to make use of a Recurrent Neural Network (Elman 1990), in which the phonological string is processed one segment at a time (see further sec. 4.2). Touretzky and Wheeler (1991) directly address the overgeneration problem noted by Pinker and Prince (1988) by developing a connectionist model that performs mappings from one string to another, and which cannot represent reversals (see also Gasser & Lee 1990 on the difficulty of reversals in Simple Recurrent Nets).

Pinker and Prince (1988) question several other aspects of the representation of past tense formation in the Rumelhart and McClelland model. The critique that most defined the course of future research was the claim that the irregular and regular past tense are the product of separate systems, rather than being produced by a single cognitive module. In Pinker and Prince’s view (p. 122-123), regularities in the irregular past tense, such as the association of particular expressions of the past tense with particular final consonants ([ei] to [ʊ] with a final [k], no change with final [t] or [d]), are “family resemblances” that are the product of the memory system, which may well be formalized in connectionist terms. That is, the irregular past tenses are lexically stored, and anything that looks like a rule-governed regularity is in fact a product of how the words are stored. The regular pattern, on the other hand, is the product of a morphosyntactic rule, or rules, that add the -ed morpheme, and phonological rules of voicing assimilation and vowel epenthesis that yield the contextually appropriate surface forms. For overviews of the subsequent research from what we might call the Pinkerian perspective, see Pinker (1999), Pinker and Ullman (2002), Marcus (2001: 68 ff.) and from the connectionist perspective, see McClelland and Patterson (2002), and Seidenberg and Plaut (2014). Debates between proponents of one and two systems models are pervasive in cognitive science; for a recent trenchant critique of two systems models of category learning, see Newell et al. (2011).

Since I have been presenting Pinker and Prince’s critique as coming from a generative perspective, I should be clear that a two systems approach is not inherent to a generative analysis of the past tense, and is probably not even standard in that tradition. Chomsky (1957: fn. 8) in fact sketches a rule-based analysis of cases of vowel change like take/took, which is developed in Chomsky and Halle (1968: p. 11) and Halle and Mohanan (1985: 104ff.), and Albright and Hayes (2003) present a quite different single system rule-based analysis. Although many phonologists might in fact believe that lexically irregular morpho-phonology is better handled by a system of lexical analogy than by a rule-based system (see Albright and Hayes for a critique of that view), it is rare to see that view developed into analyses, presumably because most phonologists do not have a formal system at hand for constructing the lexical part of the analysis. A notable exception is Bybee’s model of Natural Generative Phonology, which draws a strict distinction between productive phonological rules and semi-productive morphophonology; Bybee (1988) embraces connectionism as a means of formalizing her lexical networks (see further Bybee & McClelland 2005).5

5 As Bermúdez-Otero (2016) points out, another generative approach that seems compatible with the two-systems view is Jackendoff’s (1975) work on Lexical Redundancy Rules, and there are also relations to ideas in Lexical Phonology (Kiparsky 1982; see Kaisse & Shaw 1985; Kenstowicz 1994 for overviews). See also Liberman (2004) on the divergence between generative practice and Pinker’s two systems
The debate between connectionists and proponents of rule-based models of cognition often turns on a definition of what it means for a model to be rule-based, or connectionist. Lachter and Bever’s (1988) critique of Rumelhart and McClelland (1986), and Marcus’ (2001) critique of its connectionist successors, challenge the extent to which the models can truly be claimed to be alternatives to rule-based models. Although Rumelhart and McClelland’s model of the past tense clearly lacks the rules of a standard analysis of the regular past tense, it adopts relatively standard phonological features, and Lachter and Bever (p. 211) argue that choice, as well as the particular configurations of features for the nodes, essentially engineers a rule-based solution into the model. Marcus (p.83) argues that:

the closer the past tense models come to recapitulating the architecture of the symbolic models—by incorporating the capacity to instantiate variables with instances and to manipulate...the instances of those variables—the better they perform.

The absence of variables in the Rumelhart and McClelland (1986) model is another of the primary targets of Pinker and Prince’s (1988) critique (see p.176), and it is the presence of variables that Marcus (2001) takes as part of the definition of a “symbolic” model, and key to its success – the other part is the ability to manipulate those symbols with algebraic rules (see Smolensky 1988 on the subsymbolic nature of connectionism; and see Hummel 2010 for a recent discussion of the properties of symbolic systems).

Some of the back-and-forth in the past tense debate can be tiring precisely because it consists of one side accusing the other of not being true to its principles in incorporating aspects of the first sides’ theory. But this aspect of the debate is ultimately instructive in that it shows that the space between connectionist and generative models of language is more fluid than the rhetoric might sometimes suggest. Some points related to this fluidity have already been made above: (1) there is nothing about a connectionist model that prohibits the use of symbols, including variables, and other representations developed in linguistic traditions – though see Smolensky (1988) on why sub-symbolic representations are more natural (2) a generative rule-based model can, and often does, have the very specific rules needed to model irregular morpho-phonology (3) a generative model is not fully innatist in that parameters need to be set by experience (4) a connectionist model is not fully emergentist in that much of its structure must be specified. None of this is controversial, but it might not be apparent when each of the traditions is contrastively labeled with these characteristics.

Before moving on to discussion of contemporary models that illustrate the fruitfulness of integration across the connectionist-generativist divide, it is worth saying a few words about another model characteristic that might be taken as definitional of each of generativism and connectionism. Rumelhart and McClelland’s (1986) past tense model uses a probabilistic interpretation of a sigmoid activation function, and thus produces probabilities over different outputs for a given input. Models of generative grammar, from Chomsky (1957) onwards, typically use deterministic rules that produce a single approach, and Embick and Marantz (2005) on some of the issues in relating generative theories to experimental data on the past tense.
output for a given input. Neither of these choices are fixed however: the examples of perceptrons in section 2 are deterministic, and rules can be given a probability of application (see e.g. the probabilistic formulation of Labov’s variable rules in Cedergren & Sankoff 1974; and the probabilistic formulation of minimalist syntax in Hunter & Dyer 2013; see Lau, Clark & Lappin 2017 for recent discussion of probabilistic grammars and sentence acceptability judgments). The past tense debate also provides a reason to formalize a rule-based model probabilistically: as children acquire the regular -ed past tense, its probability of use seems to increase gradually (McClelland & Patterson 2002:467–468).\(^7\) Pinker and Prince (1988:164) in fact sketch a rule-based approach to acquisition that incorporates “competition among multiple regularities of graded strength”, which is elaborated on with explicitly stochastic rules in Albright and Hayes (2003).

In my view, rather than yielding a single victor, the past tense debate provided important lessons for the further development of both the generativist and connectionist traditions, as well as their hybrid offspring. The connectionist models of the past tense, including that of Rumelhart and McClelland, show that simple and explicit models of learning can be combined to good effect with the representational structures developed in linguistics: learning need not be over unstructured strings of words, as in the models criticized by Chomsky (1957). Perhaps more controversially, there also seem to be lessons for generativists about the graded structure of language; see for example, the discussion of the claim that “no mere mortal has ever flown out to center field” (Kim et al. 1991; Pinker 1999:chap. 6) in Seidenberg and Plaut (2014:1208–1212). The generativist critiques bring to the fore the structural complexity of language, which is not well captured by Rumelhart and McClelland’s model or its immediate successors. Perhaps more controversially, they also seem to indicate the fruitfulness of incorporating richer combinatorial structure in the representations manipulated by connectionists’ models.

4. Fusion

Despite – or perhaps because of – the debates between proponents of connectionist and generative approaches to the study of language, at the end of the 1980s and beginning of the 1990s some linguists began developing connectionist models of language, which often incorporated representational assumptions from generative linguistics (e.g. Lakoff 1988; Hare, Corina & Cottrell 1989; Legendre, Miyata & Smolensky 1990; Goldsmith 1993; Lakoff 1993; Wheeler & Touretzky 1993; Gupta & Touretzky 1994). Interest developed from the other “side” as well: some of the just cited papers were collaborations with connectionists (Smolensky, Touretzky), and connectionist research on language often began to make use of explicitly hybrid models (see e.g. many of the contributions to Sharkey 1992).\(^8\) In the following sections, I cover a small part of the work\(^9\) over the last

---

\(^6\) See the interview with Chomsky in Katz (2012) for an expression of continued skepticism about probabilistic approaches (to AI), and Norvig (n.d.) for an extended reply to earlier related comments.

\(^7\) A two systems theory might not in fact need probabilistic rules to capture these data, provided it had probabilistic lexical access that competed with rule application. As far as I know, there are no implemented versions of such an approach that compete with the connectionist accounts.

\(^8\) Rosenblatt was himself an advocate of a hybrid approach to cognition, in the sense that he saw some cognitive processes as outside of the domain of perceptron theory (Rosenblatt 1964).
30 years that synthesizes aspects of connectionism and generative linguistics: section 3.1 discusses an approach to generative grammar resulting from the importation of constraint interaction, a relatively high-level abstraction from connectionism, while section 3.2 goes on to discuss the ongoing controversy over the representation of syntactic structure in connectionist networks.

4.1 Constraint interaction in generative grammar
The title of this subsection is the subtitle of Prince and Smolensky’s (2004) book originally circulated in 1993 that introduced Optimality Theory (OT), a particularly fruitful connectionist-generativist fusion (see the now more than 1300 contributions to the Rutgers Optimality Archive – http://roa.rutgers.edu).

OT is a descendent of Chomsky’s (1981) Principles and Parameters framework in that it similarly posits a rich UG, with the goal of delimiting the space of possible languages. Instead of parameters whose values are fixed by a learner, OT has constraints whose ranking must be determined. A close relative of OT is Harmonic Grammar (HG; Legendre, Miyata & Smolensky 1990; Smolensky & Legendre 2006), which numerically weights, rather than ranks, its constraints. I will illustrate the OT/HG formalization of generative grammar using HG’s weighted constraints, since this represents a more direct fusion with connectionism (see Pater 2016a for discussion of HG/OT similarities and differences).

Prince and Smolensky (2004; ch. 4) provide an extended argument for constraint interaction in the domain of word stress, comparing the Extrametricality parameter (Hayes 1980) to a violable Nonfinality constraint. An Extrametricality parameter places word-final syllables outside of the domain of word stress, while a Nonfinality constraint penalizes stressed word-final syllables, but can be violated under the compulsion of other constraints. To see how violable constraints can produce different results from an inviolable rules or constraints, consider the interaction of Nonfinality with a Weight-to-Stress constraint that requires heavy syllables to be stressed. These constraints come into conflict when a heavy syllable is in final position. The Tableau in (6) shows a bisyllable with a heavy final syllable (one with a coda nasal) and a light initial (a CV syllable), and two candidate stress placements. The first candidate, batán, has stress on the final syllable, and thus violates Nonfinality, indicated with a negative integer in its column. The second candidate has stress on the light syllable, and thus violates Weight-to-Stress. The constraints’ weights are provided underneath their names: Weight-to-Stress has a higher weight than Nonfinality (5 vs. 2). The column labeled Harmony gives the weighted sum of constraint violations. In a deterministic version of HG, the candidate with highest Harmony, in this case bátan, is picked as the Output – the optimal candidate.

(6) A Harmonic Grammar tableau

<table>
<thead>
<tr>
<th>Input: batan</th>
<th>Weight-to-Stress</th>
<th>Nonfinality</th>
<th>Harmony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: batán</td>
<td>5</td>
<td>−1</td>
<td>−2</td>
</tr>
<tr>
<td>bátan</td>
<td>−1</td>
<td></td>
<td>−5</td>
</tr>
</tbody>
</table>

9 One particularly notable omission is research in the Gradient Symbol Systems framework (Smolensky, Goldrick & Mathis 2014), which would require another paper to cover.
The notion of constraint interaction – of one constraint overriding another – seen in this tableau can be abstractly seen in the connectionist network for XOR in section 2.1, in which the constraint against black stars activating the output neuron (negative weight on the relevant connection) outweighs the constraint that black objects and stars should activate it (positive weights on the relevant connection).

So far, we could equally analyze this situation with an Extrametricality parameter that is turned off, so that the final syllable is eligible to be stressed, and a Weight-to-Stress parameter that is turned on, so that the heavy syllable is picked over the light one. Consider, however, the following tableau, which shows a form in which there is no heavy syllable, and in which Nonfinality can be satisfied without violating Weight-to-Stress. Nonfinality prefers the candidate with stress on the initial syllable over stress on the final syllable. If Nonfinality were replaced by an inactive Extrametricality parameter, it could not be used to account for the lack of final stress in this instance. Cases like this are referred to in the OT literature as the emergence of the unmarked, or more broadly, as non-uniform constraint application.

(7) Illustration of “emergence of the unmarked”

<table>
<thead>
<tr>
<th>Input: bata</th>
<th>Weight-to-Stress</th>
<th>Nonfinality</th>
<th>Harmony</th>
</tr>
</thead>
<tbody>
<tr>
<td>batá</td>
<td>5</td>
<td>-1</td>
<td>-2</td>
</tr>
</tbody>
</table>

Output: báta

The switch from parameters to violable constraints has consequences for both the study of language typology, and of learning. For language typology, it becomes possible to maintain relatively general formulations of constraints while still accounting for details of individual languages. In parametric theory, the observation of surface violations of constraints in a language in which they are generally active leads to a range of analytic responses, including simply changing a general constraint into a set of more specific ones that can remain inviolable – this tactic is the main target of Prince and Smolensky’s (2004) attack on Extrametricality. In a review of Halle and Vergnaud’s (1987) parametric metrical theory of word stress, Dresher (1990) discusses some of its additional “extraparametric devices”, and concludes that their use draws into question Halle and Vergnaud’s claims that their theory is more typologically restrictive than its competitors. Dresher (p. 184) is in fact pessimistic that the relative restrictiveness of the theories can be determined:

At a time when all versions of metrical theory command such arsenals,

---

10 OT and HG constraints typically only assign violations, or negative values (though see Kimper 2016)
11 I am abstracting from another dimension on which OT and parameter-based theories differ. Prior to OT, constraints in generative grammar interacted with rules, rather than with one another (see McCarthy 2002; Prince & Smolensky 2004; McCarthy, Pater & Pruitt 2016 for comparison). The story of this particular marriage between connectionism and generative linguistics is largely about the quest for improved constraint-based theories of linguistics – see e.g. the comments by Alan Prince in Pater (2016b) – though it also about finding a neural grounding for symbolic systems (Smolensky & Legendre 2006). Much of the controversy around OT concerns its abandonment of serial rule ordering (Vaux & Nevins 2008).
comparisons of expressive power are likely to remain inconclusive.

The situation changed dramatically with the introduction of violable constraints (see e.g. Kager 2005 for an extended comparison of two OT theories of word stress). The greater success in constructing and comparing theories of typology was partly due to the success of the associated learning algorithms. Both OT and HG have provably convergent learning algorithms that will find a ranking, or a weighting, for a set of constraints for any set of candidate outputs that can be made jointly optimal (see Tesar & Smolensky 2000 on OT; Potts et al. 2010; Boersma & Pater 2016 on HG). Given the candidate sets and their violation profiles, we can thus determine which potential languages (sets of optima) are in fact generated by the constraint set. This learnability approach to typology calculation was pioneered by Hayes et al. (2013) for OT (see also Prince, Tesar & Merchant 2015), and has been extended to HG and serial variants of OT and HG by Staubs et al. (2010).

Studies of typology in OT and HG typically make use of deterministic variants of the theories, in which within-language variation is abstracted away from. Probabilistic variants of OT and HG have also been developed (see the survey in Coetzee & Pater 2011), and alongside them learning algorithms. Maximum Entropy Grammar (MaxEnt; Goldwater & Johnson 2003), a probabilistic variant of HG, has the distinction of having associated provably convergent learning algorithms, due to its grounding in mathematically well-understood models from other domains, including neural networks (Smolensky 1986; Johnson 2013). In MaxEnt, the probability of an output is proportional to the exponential of its Harmony. For example, with the weights in Tableau (6), batán would have probability 0.95, rather than being deterministically picked as optimal.

As well as modeling variation in final state grammars, probabilistic OT and HG grammars have also been used to model the variation in grammars in the course of acquisition (see the overview in Jarosz 2010; see also Moreton, Pater & Pertsova 2015). One class of gradual learning algorithm used in this work includes an application of Rosenblatt’s Perceptron convergence procedure (Pater 2008; Boersma & Pater 2016); this class of inter-related algorithms also includes Boersma’s (1997) widely used learner for Stochastic OT (see also Boersma & Hayes 2001) and (Stochastic) Gradient Descent for MaxEnt (Jäger 2007; Moreton, Pater & Pertsova 2015).

The just-mentioned convergence guarantees of OT and HG learning algorithms come with an important caveat: they apply only when the structure of the learning data is supplied in whole – when all the constraint violations of each learning datum are known. To continue with the stress example, finally stressed batán might be analyzed as having a trochaic (left-headed) foot on the final syllable, or an iambic (right-headed) foot that parses both syllables. As shown in (8), each representation incurs distinct constraint violations, since in one case the initial syllable is unparsed, violating Parse-Syllable, while the other falls afoul of Trochee, a constraint demanding left-headed feet. If we supply the learner with only the overt form, it must choose between the two full parses (or assign probability to them).

(8) An example of a hidden structure problem

<table>
<thead>
<tr>
<th>Overt form</th>
<th>Full structure</th>
<th>Parse-Syllable</th>
<th>Trochee</th>
</tr>
</thead>
<tbody>
<tr>
<td>batán</td>
<td>ba(tán)</td>
<td>−1</td>
<td></td>
</tr>
</tbody>
</table>
While supplying full structures might be unproblematic in typology calculation or in modeling some cases of variation, there are many cases of linguistic analysis in which one might not be committed to a particular full structure for each piece of data, and would like a learner to find an appropriate grammar. In addition, dropping the idealization of full access to structure is part of moving to a more realistic model of human language acquisition.

Learning with hidden structure (AKA learning with structural ambiguity) is generally accomplished in OT and HG by using the current state of the grammar to pick amongst the full structures for a piece of learning data (Tesar & Smolensky 2000), or to assign a probability distribution to them (Jarosz 2013; Boersma & Pater 2016; Boersma & van Leussen 2017). A MaxEnt version of this general approach (e.g. Pater et al. 2012; Johnson et al. 2015) creates a single vector (row) of constraint scores for a partially structured learning datum by summing over the probability weighted vectors of all of the corresponding full structures (see Staubs & Pater 2016; Nazarov & Pater 2017 for extensions to serial variants of MaxEnt). None of these methods is guaranteed to converge on an optimal grammar, and their development and comparison is an area of ongoing research (as Boersma 2003 points out, non-convergence can be an advantage, insofar as it corresponds to human learning difficulties or typological gaps).

As was already noted in the discussion of gradual learning algorithms, the learning methods for MaxEnt models and neural networks are highly overlapping. Both can be learned with gradient-based optimization methods, including Gradient Descent (recall that the gradient indicates the direction of weight change). When the gradient for a neural net with one or more hidden layers is constructed using backpropagation, or when a gradient for a MaxEnt model is constructed with hidden structure (see Staubs & Pater 2013 for a derivation of the gradient), there is no guarantee that these methods will find the best set of weights for the model, in terms of optimizing the fit of the model’s predictions to the data. That is, the learner may not find the global optimum, and may instead be trapped in a local minimum of error. Because of the current prominence of deep learning in AI, there is substantial research effort being expended to improve the performance of learners for multi-layer perceptrons, and many of the proposals can be adapted directly to MaxEnt (see e.g. Neelakantan et al. 2015).

Learners for parametric models are also subject to local optima because of structural ambiguity (Gibson & Wexler 1994). A typical response to this problem, in Gibson and Wexler and elsewhere, is to search for triggers (or cues) in the learning data that unambiguously correspond to the non-default value of each parameter, and then build those into the learner; one might also stipulate an ordering on the setting of the parameters (e.g. Dresher 1999). It is generally an advantage of violable constraints that such additional machinery is unnecessary (Tesar & Smolensky 2000), but recent work by Gould (2015) and Nazarov and Jarosz (2017) indicates that probabilistic parametric learners may also succeed without triggers.\(^\text{12}\)

\(^\text{12}\) Nazarov and Jarosz (2017) find that Yang’s (2002) earlier trigger-free probabilistic learner succeeds at learning only 1 of 23 languages generated from a set of metrical parameters; their own learner succeeds on 22 of them. It is also worth noting that Tesar (2004) presents a learner that can provably cope with
One particularly interesting application of MaxEnt to the learning of phonology falls somewhat outside of standard OT/HG frameworks, in two respects. Hayes and Wilson (2008) develop a model of phonotactics that defines a probability distribution over the space of possible words. Unlike standard OT/HG as presented above, there is no mapping from an Input an Output (or to put it differently, the Input is an undifferentiated “word”, rather than a word of a particular phonological shape). Also unlike standard OT/HG, the constraint set is not taken as a given (see Hayes & Wilson 2008:425; Moreton & Pater 2011 for discussion of the consequences for study of typology). Hayes and Wilson propose a method for constraint induction that chooses amongst candidate constraints according to a set of heuristics (see Wilson & Gallagher 2016 for an alternative gain-based approach and comparison with other theories; see Berent et al. 2012 on evidence for the incorporation of variables). The hidden layer of a neural net allows for an alternative to constraint induction – which was hinted at in section 2.1, where we saw a hidden layer node sensitive to feature conjunction – and current research is exploring this alternative (Alderete, Tupper & Frisch 2013; Doucette 2017).

4.2 Can recurrent neural networks learn syntax?

Recurrent neural networks (RNNs; Elman 1990; Elman 1991) do perhaps surprisingly well in capturing some aspects, including long-distance dependencies, of natural language syntax, and have recently undergone a resurgence of popularity in AI applications of neural networks to language (see Goldberg 2016; Goldberg 2017 for tutorial overviews). It is unclear, however, whether they can fully represent syntactic regularities without the incorporation of hierarchical representations of the type used in generative linguistics, or alternative linguistic structure like dependency marking. In this section, I briefly introduce RNNs, provide an overview of some of the research on the learning of syntax using them, and point to some outstanding research questions.

When applied to a string of elements, such as letters, phones, or words, RNNs process one at a time, starting at one edge of the string – typically the left. The network is used to predict the next element in the sequence, and the weights are updated based on that prediction (updates can also be made on more global predictions). As illustrated in Figure 3, when moving onto to the next element in a sequence, the current hidden layer is copied as a context layer to provide an extra set of inputs to the next computation of the hidden layer activations, and the Output. The representation encoded in this copied hidden layer provides a basis for the prediction of upcoming elements based on those encountered earlier – i.e. it is a type of sequential memory.

---

structural ambiguity; it works only with deterministic OT (or HG), and cannot be used to model learning paths.
Elman (1990; 1991) uses toy language examples to show that these networks can capture some basic aspects of syntax. Elman (1990) focuses on the formation of categories (Noun and Verb), and on word order restrictions in single clauses. Elman (1991) goes on to examine RNNs’ abilities to capture dependencies across embedded clauses, finding that their predictions are in fact influenced by information retained from across the embedding.

Elman (1991:221) recognizes the preliminary nature of his results, and given the limitations of the datasets he was working with, one could reasonably either be enthusiastic about the potential for further development of this approach, or skeptical. And presumably, that degree of skepticism could well be influenced by the strength of ones’ prior belief in the necessity of a rich UG. An RNN of this type is – relatively speaking – a very impoverished theory of UG. Not only does it lack parameters or constraints encoding the substance of possible grammatical systems, the basic structures, such as syntactic categories and trees, are themselves absent, at least prior to learning. A hidden layer can form abstract representations of the data, and there are some hints in Elman’s results that those representations may do the work of explicit categories and constituent structure, but much research remains to be done, even today, to determine the extent to which they can.

Fitz and Chang (2017) provide a particularly useful overview of one line of research in this domain, and some intriguing new results. A much-discussed case of the poverty of the stimulus is auxiliary inversion in English yes-no question formation (Chomsky 1975; Chomsky 1980). The fronted auxiliary in “Is the bottle that the man is holding broken?” is displaced from the underlined verb phrase of the main clause in “The bottle that the man is holding is broken”, rather than from the embedded clause. This is a structure dependent rule, as opposed to a putative rule of “front the first auxiliary”, which would work for mono-clausal cases, but would yield the ungrammatical *“Is the bottle that the man holding is broken?” for this and other similar cases of embedding. Fitz and Chang show that an RNN that is simply trained to predict upcoming words generalizes the incorrect rule (contra Lewis and Elman 2001), but that if the learner is given the task of generating sentences given a meaning representation, it does yield the correct generalization from monoclausal training instances of auxiliary inversion to the structure dependent rule. Since they incorporate explicit propositional structure into their meaning representations, Fitz and Chang’s work can be seen as an instance of the “fusion” that is the focus of this section – they in fact see their proposal as bridging the usually antagonistic emergentist/constructivist and innatist approaches to aux-inversion (p. 243).
Whether or not Fitz and Chang’s approach will generalize to other phenomena is very much an open question. One set of challenges for RNN approaches to syntax appears in Frank, Mathis and Badecker’s (2013) study of anaphora resolution. The pair of sentences from their study (p. 197) in (9) illustrates part of the phenomenon.

(9)   a. Alice who Mary loves admires herself
     b. Alice who loves Mary admires herself

The question is who “herself” refers to. As Frank et al. point out, native speaker judgments, as well as online processing measures (Xiang, Dillon & Phillips 2009), show that it can only be Alice – not Mary, or anyone else. This is captured in grammatical accounts in terms of the positions of Alice and Mary relative to “herself” in a hierarchical structure. Frank et al. explore an RNN similar to that illustrated in Figure 3, except that the context layer is sandwiched between two hidden layers. They train it to predict upcoming words, and then use the trained network to assign interpretations by examining which nodes “herself” activates, where Alice, Mary and Sue each have a node. They find that the network correctly assigns almost all of the probability to Alice in (9a). In (9b.), however, it assigns probability of 0.16 to Mary. They diagnose the success on (9a.) as indicating RNN’s ability to capture structural relationships (though see further below), and the partial failure on (9b.) as indicating their ability to capture incorrect linear relationships: the substring “Mary admires herself” can be interpreted in other contexts with herself referring to Mary. Importantly, they argue that the sequence-based errors of networks in this task are different from those observed in human sentence processing (p. 200-201), which show sensitivity to structure.

Neural networks – especially RNNs – without prespecified linguistic structure have recently been broadly applied in AI language tasks, with considerable success. In machine translation, RNNs are used to map from a sequence of words in one language to a sequence in another, without any intermediate explicitly encoded linguistic structure (Sutskever, Vinyals & Le 2014; Bahdanau, Cho & Bengio 2016; Wu et al. 2016). These models achieve state of the art performance, doing as well, and sometimes better, than earlier models that map to intermediate levels with linguistic structure like phrases, and are the basis of useful applications like Google Translate. This “end-to-end” approach has also been applied in speech recognition, mapping from acoustic signal to text without an explicit intermediate phone layer (e.g. Amodei et al. 2016).

The success of modern RNNs in applied language tasks is due to advances in their architecture, as well as in training methods and computational hardware which jointly allow for training of large networks (e.g. very “deep” ones with many layers) on large datasets. Do these advances lead to models that can successfully represent natural language syntax without explicitly specified linguistic structure? Some reason for optimism comes from the success of modern RNNs in the modeling of sentence acceptability judgments in Lau et al. (2017). Linzen et al. (2016) explore the ability of modern RNNs to learn subject-verb agreement, and find results that are somewhat parallel to those of Frank et al. (2013): the models do have a degree of success on even
long-distance dependencies, indicating that they have learned something akin to a structural analysis, but they also have a tendency to extract incorrect linear regularities.\(^\text{13}\)

The sequence-to-sequence models used in machine translation allow for straightforward applications to the kinds of within-language mappings between representations studied by linguists. For example, Kirov (2017) trains a standard RNN-based machine translation model on the same English present-to-past tense mappings as Albright and Hayes’ (2003) stochastic rule-based model. He finds that its predictions provide a far better match to native speaker formations of novel past tense forms (“wug” test data), especially amongst irregulars, than Albright and Hayes’ own model. They can also be used to model syntactic transformations, as shown by Frank and Mathis (2007) using an earlier generation of RNN, and ongoing research is addressing the extent to which these models cope with the auxiliary inversion poverty of the stimulus problem discussed above (Robert Frank and Tal Linzen p.c.).

Current research in AI also provides some grounds for skepticism about the eventual success of “vanilla” RNNs as models of linguistic knowledge.\(^\text{14}\) There are a number of recent proposals that incorporate compositional and hierarchical linguistic structures into neural networks with the aim of improving their performance on AI tasks (e.g. Andreas & Ghahramani 2013; Socher et al. 2013; Bowman et al. 2015; Dyer et al. 2016; Yogatama et al. 2016). Alongside these proposals are observations about the fragility of models that eschew linguistic structure when they are tested on linguistically challenging data (e.g. Ettinger et al. 2017; Jia & Liang 2017). An overview of the presentations at the 2017 meeting of the Association for Computational Linguistics in fact dubbed one of the main trends as “Linguistic Structure is Back” (See 2017).

Besides testing the performance of trained neural networks, one can inspect the values of connection weights and the activation patterns produced for particular inputs to gain insight into the representations they have constructed. Elman (1990; 1991) pioneered this approach with his early RNN studies, finding evidence for the representation of syntactic categories in the hidden layer. In their study of anaphora resolution, Frank et al. (2013) conclude that the representations are not sufficiently abstract, being too tied to particular words rather than to categories. This approach has also recently been applied in speech, finding evidence for phone categories being represented in a hidden layer of a network trained to map from acoustic signals to images (Alishahi, Barking & Chrupala 2017). A particularly relevant recent result in this vein with respect to the question of how syntax may be represented in RNNs is presented in Palangi et al. (2017), who introduce Tensor Product Recurrent Networks, and show that their internal representations can be interpreted syntactically. Like the Hayes and Wilson model discussed at the end of the last section, this model occupies an interesting middle ground between the poles of innatism and emergentism, since it is given the structural building blocks of symbols and roles, but must learn their configurations.

\(^{13}\) Adger (2017) takes the failures of the Linzen et al. (2016) model as vindication of Chomsky’s (1957) arguments against statistical models of language. One could equally take its successes as vindication of a statistical approach. Both of these views are likely correct.

\(^{14}\) See also the discussion of the inability of “plain” RNNs to represent copy mappings that apply to strings outside of the training set in Marcus (2001:chap. 3) and Tupper and Shahriari (2016); this poses a problem for representing natural language reduplication. Perhaps unsurprisingly, standard sequence-to-sequence models do not seem to acquire general reduplicative mapping functions (Brandon Prickett p.c.).
5. Conclusions
When viewed from a sufficient distance, neural network and generative linguistic approaches to cognition overlap considerably: they both aim to provide formally explicit accounts of the mental structures underlying cognitive processes, and they both aim to explain how those structures are learned. When viewed more closely, especially with respect to the research practices within each tradition, they may seem to diverge sharply, with the bulk of connectionist practice involving computational learning simulation allied with AI tasks or with psychological experimentation (see e.g. papers in Christiansen & Chater 2001), and the bulk of generative practice involving grammatical analysis of linguistic systems. At a middle depth of field, one can find a growing research territory in which the bodies of knowledge and the models developed in each of these traditions are jointly applicable. In this paper, I have focused on the question of how systems that adequately represent linguistic knowledge can be learned. Some of the most promising avenues for answering that question build both on generative insights into the nature of linguistic knowledge, and on connectionist insights into the nature of learning.

References


Cherry, Colin. 1957. On human communication; a review, a survey, and a criticism.


Hayes, Bruce, Bruce Tesar & Kie Zuraw. 2013. OTSoft. Los Angeles: UCLA.


Rosenblatt, Frank. 1957. The perceptron, a perceiving and recognizing automaton (Project PARA). Cornell Aeronautical Laboratory.


