COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science (2015) 1–28

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/cogs.12258

Learning Orthographic Structure With Sequential Generative Neural Networks

Alberto Testolin,[a,b] Ivilin Stoianov,[b,c] Alessandro Sperduti,[d] Marco Zorzi[b,e,f]

[a]*Department of Developmental Psychology and Socialisation, University of Padova*
[b]*Department of General Psychology, University of Padova*
[c]*Cognitive Psychology Laboratory, CNRS & Aix-Marseille University*
[d]*Department of Mathematics, University of Padova*
[e]*Center for Cognitive Neuroscience, University of Padova*
[f]*IRCCS San Camillo Neurorehabilitation Hospital*

Received 22 November 2013; received in revised form 21 November 2014; accepted 2 February 2015

Abstract

Learning the structure of event sequences is a ubiquitous problem in cognition and particularly in language. One possible solution is to learn a probabilistic generative model of sequences that allows making predictions about upcoming events. Though appealing from a neurobiological standpoint, this approach is typically not pursued in connectionist modeling. Here, we investigated a sequential version of the restricted Boltzmann machine (RBM), a stochastic recurrent neural network that extracts high-order structure from sensory data through unsupervised generative learning and can encode contextual information in the form of internal, distributed representations. We assessed whether this type of network can extract the orthographic structure of English monosyllables by learning a generative model of the letter sequences forming a word training corpus. We show that the network learned an accurate probabilistic model of English graphotactics, which can be used to make predictions about the letter following a given context as well as to autonomously generate high-quality pseudowords. The model was compared to an extended version of simple recurrent networks, augmented with a stochastic process that allows autonomous generation of sequences, and to non-connectionist probabilistic models (*n*-grams and hidden Markov models). We conclude that sequential RBMs and stochastic simple recurrent networks are promising candidates for modeling cognition in the temporal domain.

Keywords: Connectionist modeling; Recurrent neural networks; Restricted Boltzmann machines; Probabilistic graphical models; Generative models; Unsupervised learning; Statistical sequence learning; Orthographic structure

Correspondence should be sent to Marco Zorzi, Department of General Psychology, University of Padova, Via Venezia 12, Padova 35131, Italy. E-mail: marco.zorzi@unipd.it

1. Introduction

A growing body of research suggests that the ability to extract statistical regularities from the environment is a powerful and general learning mechanism of the brain, which operates across domains, modalities, and development (see Krogh, Vlach, & Johnson, 2013; for review). For example, a seminal developmental study (Saffran, Aslin, & Newport, 1996) showed that infants can efficiently exploit statistical relationships between neighboring speech sounds to segment words from fluent speech. These results highlight the prominent role of statistical learning in language acquisition, thereby suggesting the importance of experience-dependent mechanisms for extracting useful structure from the auditory stream (see Romberg & Saffran, 2010; for review). In the same vein, statistical regularities can be learned from visual stimuli in an unsupervised way (Fiser & Aslin, 2001). Statistical learning over speech streams has also been reported in rodents (Toro & Trobalón, 2005) and non-human primates (Hauser, Newport, & Aslin, 2001), and it has been recently shown that baboons can successfully exploit statistical regularities even for discriminating well-formed from ill-formed visual words (Grainger, Dufau, Montant, Ziegler, & Fagot, 2012).

However, whether statistical properties are sufficient to fully develop high-level cognitive abilities, like those involved in human language that encompass learning of abstract, "rule-based" constructs, is still debated (e.g., Marcus, 1999; McClelland & Plaut, 1999; Peña, Bonatti, Nespor, & Mehler, 2002; Perruchet, Tyler, Galland, & Peereman, 2004; Seidenberg, MacDonald, & Saffran, 2002). This debate is also reflected in modeling approaches that place different emphasis on explicit representations in processing of structured information (Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010; McClelland et al., 2010). On the one hand, structured Bayesian models exploit explicit (though flexible) forms of knowledge representation, which can be shaped and refined by learning processes that operate according to some inductive biases. In contrast, the emergentist approach focuses on the underlying mechanisms that could produce the observed phenomena: There is no need to postulate a specific hypothesis space, because the knowledge of the system is essentially implicit and it emerges in the model behavior as a result of the learning process. As a prominent example of the latter approach, connectionist models provide a biologically inspired way to study implicit learning and knowledge representation, by proposing how cognition might emerge from the neural substrate through distributed processes that operate over large networks of simple, neuron-like interconnected units (Rumelhart & McClelland, 1986).

Neural networks allow researchers to simulate both skilled performance and breakdowns caused by brain damage, thereby providing a unique way to study how dysfunctions of the underlying neural processes affect cognition and behavior (e.g., Glasspool, Shallice, & Cipolotti, 2006; Hinton & Shallice, 1991; Perry, Ziegler, & Zorzi, 2007; Plaut, McClelland, Seidenberg, & Patterson, 1996). Connectionist systems gradually learn from examples, thereby also allowing for exploration of developmental trajectories (Elman et al., 1996). Learning in artificial neural networks can be cast within the mathematical framework of statistical learning theory (Jain, Duin, & Mao, 2000; Jordan &

Sejnowski, 2001; Neal, 1995), thus suggesting the adequacy of these models to investigate the role of statistical information for the development of complex abilities. Although several attempts have been made to merge structured and connectionist models (e.g., Marcus, 2003; Smolensky, 2006), we still lack a comprehensive theory of how high-level cognition could be implemented in neural circuits. Another crucial modeling issue is how to deal with the temporal dimension, which is ubiquitous in cognition (and, particularly, in linguistic processes), because the temporal structure must be found in time by extracting correlations between elements arranged into a sequential input stream (Elman, 1990). Learning of time dependencies has proven to be a difficult problem for neural network models (Bengio, Simard, & Frasconi, 1994; Servan-Schreiber, Cleeremans, & McClelland, 1991), and it is still an intensively researched topic (Martens & Sutskever, 2011; Sutskever, 2013).

In this study, we tackle the issue of learning sequences of elements within the framework of probabilistic generative models (Hinton & Ghahramani, 1997; Rao, Olshausen, & Lewicki, 2002), which have recently attracted much interest in both the machine learning and cognitive science communities. The fundamental hypothesis behind this approach is that the brain (and cortical circuits in particular) progressively learns an internal model of the world from sensory information and actively uses such acquired knowledge to infer causes and make predictions about relevant events (Clark, 2013; Dayan, Hinton, Neal, & Zemel, 1995; Friston, 2005; Hinton & Ghahramani, 1997). Perception can thus be formulated as probabilistic inference on input data, given a set of hidden causes that have been learned from the statistical regularities inherent in the natural world. This Bayesian formulation deals with ambiguity of sensory input and with the intrinsic uncertainty of environmental dynamics, and also provides a coherent theory about how learning can integrate new evidence to refine beliefs of the model. This perspective posits a critical role of unsupervised learning to build such internal representations: There is no need to have an additional external signal that guides learning, as the aim is to reproduce incoming information as accurately as possible by discovering its hidden causes. This approach is also appealing from a neurobiological perspective, because it could offer a unified account of perception and action, explain the functional role of attention, and capture the special contribution of cortical processing to adaptive success (Clark, 2013). The prediction is that patterns of neural activity in a high-level area must not only represent the data; they must also be capable of generating patterns of activity at earlier sensory stages that resemble the activity evoked by the external world (Abbott, 2008), in line with the idea of predictive coding (Huang & Rao, 2011). Notably, neural signatures of model-driven visual perception have been found in the V1 cortical activity of awake ferrets (Berkes, Orbán, Lengyel, & Fiser, 2011). Also human neuroimaging data suggest that expectations facilitate perceptual inference in a noisy and ambiguous visual task by sharpening early sensory representations (Kok, Jehee, & de Lange, 2012).

Generative models can be implemented in stochastic recurrent neural networks that learn to reconstruct the sensory input (i.e., maximum-likelihood learning) through feedback connections and Hebbian-like learning mechanisms, such as in the Restricted Boltzmann Machine (RBM; Hinton, 2002; Smolensky, 1986). RBMs can also be used as building blocks to learn hierarchical generative models, also known as "deep networks"

(Hinton & Salakhutdinov, 2006), in which increasingly complex and structured representations emerge as a function of network depth. Although deep learning has proven to be very successful and it is a hot research topic among the machine learning community (Bengio, 2009; Bengio, Courville, & Vincent, 2012), its potential has not yet been extensively explored by cognitive scientists (see Zorzi, Testolin, & Stoianov, 2013; for review and discussion). Finally, RBMs (and, consequently, deep networks) can be easily implemented on parallel computing architectures (Testolin, Stoianov, De Filippo De Grazia, & Zorzi, 2013), thereby allowing to efficiently train large-scale models (Dean et al., 2012; Raina, Madhavan, & Ng, 2009).

Here, we addressed a problem that occurs in the temporal domain, which consists of learning statistical constraints on the structure of event sequences. We focus on written language processing, but rather than investigating learning of syntactic structure from sequences of words (e.g., Bengio, Ducharme, Vincent, & Jauvin, 2003; Mnih & Hinton, 2007; Sutskever, Martens, & Hinton, 2011) we investigated the simpler problem of learning orthographic structure from sequences of letters that form the words of a language. The orthographic structure (i.e., the transition probabilities between letters) that guides the construction of legal words, typically described as "graphotactic rules," had to be inferred from a corpus of English monosyllables. Traditional connectionist modeling approaches for learning temporal structure exploit the well-known simple recurrent networks (SRNs; Cleeremans & McClelland, 1991; Elman, 1990; Nerbonne & Stoianov, 2004; Stoianov, 2001), possibly extending them to allow an internal encoding of sequences (Pollack, 1990; Sibley, Kello, Plaut, & Elman, 2008; Stoianov, 1999, 2001). However, state-of-the-art performance in sequence learning is often achieved using probabilistic models, such as Hidden Markov Models (HMMs; Rabiner, 1989), that are not formulated according to the principles of neural computation (O'Reilly, 1998) and therefore do not explain how sequence learning could be carried out in a neuronal architecture. We investigate whether the statistical structure that is implicitly contained in letter sequences can be learned by a recently proposed extension of the RBM that can deal with temporal data, known as the Recurrent Temporal Restricted Boltzmann Machine (RTRBM; Sutskever, Hinton, & Taylor, 2008). To our knowledge, this is the first attempt to model character-level sequential processing exploiting a generative neural network. Hinton and colleagues tested the RTRBM on simple video sequences consisting of three balls bouncing in a box, demonstrating that the network can be successfully applied in a temporal domain, which implies smooth dynamics. Here, we further explore the capability of temporal RBMs by modeling sequences that imply transitions between discrete (i.e., symbolic) elements, as typically found in the printed language domain.

We note that, although RTRBMs and SRNs have some common characteristics, they differ in several fundamental aspects. In SRNs, supervised learning is used to establish a mapping between the input (i.e., the current element of the sequence plus contextual information) and a separate output representation (i.e., the prediction of the following element), as illustrated in Fig. 1C. In contrast, RTRBMs use a common layer for encoding both the input and the model's prediction (see Fig. 1A), and learn to process sequential information in a completely unsupervised way by trying to accurately reproduce (i.e.,

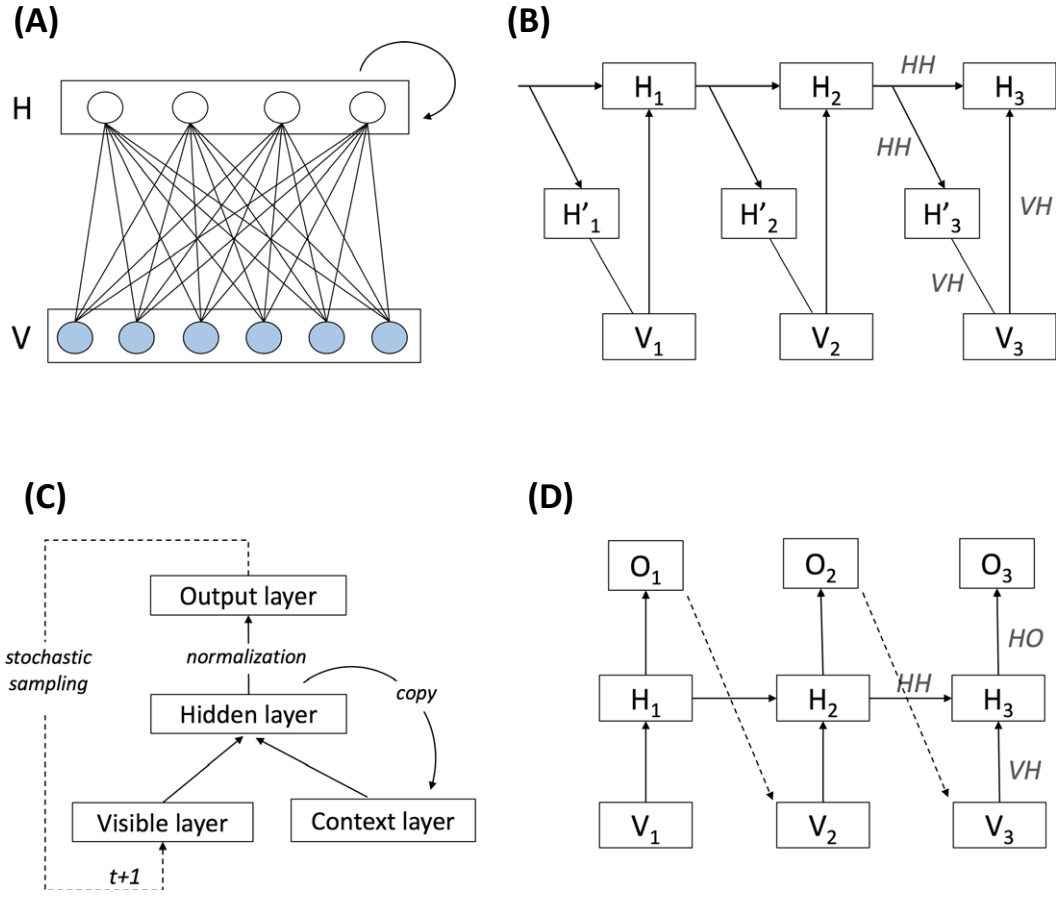

Fig. 1. (Panel A) Graphical representation of a Restricted Boltzmann Machine. Units in the visible layer *V* are fully connected with units in the hidden layer *H*, but there are no within-layer connections. The additional set of hidden-to-hidden delayed connections (curved arrow) allows extending the basic architecture to obtain the Recurrent Temporal Restricted Boltzmann Machine (RTRBM). (Panel B) Schematic diagram of the RTRBM processing a three-element sequence (from left to right). Note that there is one hidden layer with real-valued activations (H) that is used for inference and one with binary activations (H') that is used during the generative phase. The weights parameterization is reported for the last element, where visible-to-hidden connections are indicated with VH and hidden-to-hidden connections are indicated with HH (adapted from Sutskever et al., 2008). (Panel C) Graphical representation of the Stochastic Simple Recurrent Network (SSRN). The output activations are first normalized, and an external stochastic process then samples the next element of the sequence, which is given as input to the network at the following time step. (Panel D) Schematic diagram of the SSRN generating a three-element sequence. Note that this architecture requires an additional set of hidden-to-output (HO) weights.

generate) the training sequences. Processing in SRNs is thereby inherently deterministic and essentially input-driven (i.e., bottom–up), whereas RTRBMs can autonomously produce top–down activations on the sensory units from internal representations through

their intrinsically stochastic dynamics, as explained below. As the standard formulation of SRNs cannot be used to generate sequences, we propose an extension of the basic SRN in which an external stochastic process is used to sample the next element following a given sequence, according to learned conditional probabilities.

Our findings show that both the RTRBM and the extended, stochastic SRN (henceforth, SSRN) can successfully learn the orthographic structure of English words, by building a probabilistic model of letter sequences that can be used to predict the next letter given a certain context, as well as to autonomously generate high-quality (i.e., graphotactically correct) pseudowords. We compared the prediction performance of the connectionist models with that of other probabilistic models, that is *n*-grams and HMMs. We also evaluated the generative ability of the considered models, in terms of quality of the letter strings produced in comparison to existing pseudoword generators. Finally, we discuss the potential of sequential generative neural networks for modeling more complex structure in the temporal domain, and we highlight some open questions that should be addressed to further explore this promising area of research.

2. Learning temporal structure with sequential restricted Boltzmann machines

Boltzmann machines are stochastic networks of symmetrically connected, neuron-like units (Ackley, Hinton, & Sejnowski, 1985). Input patterns are given through a layer of visible units, and a separate layer of hidden units is used to model the latent causes of the data by capturing high-order statistics from the activation of visible units. A Restricted Boltzmann Machine (RBM) is obtained by removing all the within-layer lateral connections from the network, which therefore becomes a bipartite graph (as shown in Fig. 1A). The dynamics of the network is governed by an energy function that describes which configurations of the units are more likely to occur by assigning them a joint probability value:

$$P(v,h) = \frac{e^{-E(v,h)}}{Z},$$

where v represents visible units, h represents hidden units, and Z is a normalizing factor known as a partition function. The energy function E is defined as:

$$E(v,h) = -b^{\mathrm{T}}v - c^{\mathrm{T}}h - h^{\mathrm{T}}Wv,$$

where W is the matrix of connection weights, b and c are the biases of visible and hidden units, respectively, and T indicates the transpose operator. The learning process gradually changes weights and biases to minimize the discrepancy between the data distribution and the model distribution, that is, the objective of learning is to construct an internal generative model that produces examples with the same probability distribution as the examples contained in the training dataset. Unfortunately, computing the model's

expectations requires iteratively performing block Gibbs sampling until the network reaches equilibrium (Ackley et al., 1985). The high computational demand of this procedure has long been a strong limitation of the original learning algorithm, thereby limiting the practical use of this type of models. However, a recent efficient learning procedure, called contrastive divergence (CD) (Hinton, 2002), greatly speeds up the learning process, thereby allowing fast training of large RBMs. Notably, once an RBM has learned a good generative model of the training data, its internal representations (i.e., the activations of its hidden units) can be used as input to another RBM, thereby building a "deep" network that learns a hierarchical generative model (Hinton, 2007).

This modeling framework has been successfully applied to many perceptual tasks (see Zorzi et al., 2013; for a tutorial review), but its use has been limited to the modeling of learning tasks that imply static input patterns. In the case of temporal data, input patterns are not independent from each other because they are supplied in a precise sequential order. A generative model should therefore consider not only the current sensory evidence (i.e., visible units activations) but also the history provided by the previously presented items of the sequence. In particular, the aim is to predict the probability distribution of an element of a sequence, possibly given other preceding elements as context. The RTRBM (Sutskever et al., 2008) extends the architecture of traditional RBMs by adding a set of recurrent connections in the hidden layer, which are used to propagate information over time to keep track of past states of the system. This augmented network can be seen as a partially directed graphical model (see Fig. 1B), where some of the parameters are not free but are instead parameterized functions of conditioning random variables (i.e., the context).

Probabilistic models of sequential data with hidden variables, which act as an internal state, can capture the temporal dependencies between elements of a sequence by using either a localist state representation (as in HMMs) or by exploiting a distributed representation of the state (as in RTRBMs and SRNs). In the latter case, each entity is represented by a pattern of activity distributed over many hidden units, and each unit is involved in representing many different entities (Hinton, McClelland, & Rumelhart, 1986). Moreover, unlike models that use slot-based representations on which visible units encode position-specific elements of a sequence, recurrent neural networks learn to gradually integrate temporal information over time, generalizing knowledge about letters across positions by encoding their statistical relations in the hidden layer. In this way, the internal representations created in the hidden layer can implicitly encode distal temporal interactions that can span an arbitrary number of elements. The network might therefore in principle be able to build fixed-width, internal representations of whole sequences as static activation patterns (Sibley et al., 2008; Stoianov, 1999). In this work, we focused on learning orthographic structure, and the systematic investigation of internal representations in RTRBMs is left for future studies (see Testolin, Sperduti, Stoianov, & Zorzi, 2012, for a preliminary report).

The joint distribution of a whole sequence of T visible and hidden variables $(v_1^T, h_1^T) = \langle (v_1, h_1) \ldots (v_T, h_T) \rangle$ induced by an RTRBM is defined as:

$$P(v_1^T, h_1^T) = P_0(v_1)P(h_1|v_1)\prod_{t=2}^{T} P(v_t|h_{t-1})P(h_t|v_t, h_{t-1}),$$

where $P_0(v_1)P(h_1|v_1)$ specifies the distribution of the first pair of the sequence when no context is available. In this case, the probability distribution of visible units is not conditioned (there is no context), whereas the probability distribution of hidden units is conditioned to the state of visible units, which represents the current evidence. The following conditional distributions $P(v_t|h_{t-1})P(h_t|v_t, h_{t-1})$ are computed step by step, conditioning the visible activations v_t on the previous hidden activations h_{t-1} (contextual information) and conditioning the hidden activations h_t on both the previous hidden activations h_{t-1} and current visible activations v_t. The joint distribution of visible and hidden variables for the whole sequence is given by the product of all these conditional distributions.

During the processing of a sequence, to predict the successive visible layer activations v_{t+1} we first infer the hidden state h_t given the current element of the sequence v_t and the previous hidden state h_{t-1} using a mean field approximation (Peterson & Anderson, 1987):

$$P(h_t|v_t, h_{t-1}) = sigm(VH^{\mathrm{T}}v_t + b_h + HH\, h_{t-1})$$

where VH is the matrix of visible-to-hidden connections, b_h is the static hidden unit bias, HH is the matrix of the additional hidden-to-hidden connections, and $sigm$ is the logistic activation function:

$$sigm(z) = \frac{1}{1 + e^{-z}}.$$

The term $HH\, h_{t-1}$ represents the dynamic hidden bias, which is used to propagate contextual information over time. Once the conditional hidden activations h_t have been inferred, we can generate a prediction of v_{t+1} by starting from a random binary state of the pair (v_{t+1}, h'_{t+1}) and performing iterative block Gibbs sampling until convergence, in which the activation of the hidden units also accounts for the dynamic bias $HH\, h_t$ (see Fig. 1B). If we do not condition the model on a given context, we can let the network generate a sequence by starting from an initial learned bias and sequentially generating visible and hidden states. As discussed above, the RTRBM uses the same set of connections to perceive a stimulus and generate predictions for the upcoming stimulus (see Fig. 1B). In our simulations, the network was used in either an "inference" or a "generative" modality, but never in a mixed mode. During inference, all the visible units are hard-clamped with the current element of the sequence and the activation of the hidden units at a given time step is inferred through a single bottom–up pass (using the mean field equation reported above), considering the activation of the visible units at the current time step and the activation of the hidden units at the previous step. During generation, the bias of the hidden units is dynamically adjusted according to the previous

hidden unit activations, and then Gibbs sampling is performed in the hidden and visible layers by starting from a random initialization, until equilibrium is reached. In generation, therefore, no clamping is used in the visible layer.

It should be noted that the network processes temporal information in a strictly sequential way, one element at a time and only using the last hidden activations as context. Thus, in contrast to other probabilistic language models that introduce additional temporal connections between preceding elements and the hidden state (e.g., Mnih & Hinton, 2007), the RTRBM only exploits local temporal interactions, which can nonetheless allow to encode in the hidden layer an arbitrary number of preceding elements as context. As for RBMs, RTRBMs can be efficiently trained in an unsupervised fashion by using CD to compute the local gradient of the prediction error for each element of a sequence. The gradients are then propagated to previous time steps using backpropagation through time (for details, see Sutskever et al., 2008). In the original work, the network learned a continuous dynamic model that described the physical behavior of bouncing balls in a constrained space. Here, we address the intriguing question of whether the same network can also learn discrete dynamics, or temporal structures like the grammars describing various linguistic phenomena.

We focused on the sublexical level, investigating whether RTRBMs could exploit unsupervised learning to extract the compositional rules of elementary units forming words. These units could be either letters or phonemes, and the corresponding grammars are known as graphotactics and phonotactics (Nerbonne & Stoianov, 2004). These two grammars are strongly related because there is a tight link between letters and phonemes in alphabetic languages. In English, graphotactics and phonotactics are comparable in terms of overall complexity: There are fewer graphemic tokens than phonemic ones, but the graphotactic compositional rules are generally deeper than the phonotactic ones (Chomsky, 1970). Although most sequential connectionist models traditionally focused on spoken language (see, e.g., Elman, 1990; McClelland & Elman, 1986), our study on learning graphotactics from printed letter sequences is intended as a testbed problem for the general task of learning the structure of a sequence of discrete units. Moreover, several connectionist models of visual word recognition and reading aloud entail a serial processing mechanism (Perry et al., 2007; Plaut, 1999; Sibley et al., 2008). Most notably, sequential processing of letters is prominent during reading acquisition in childhood, whereby phonological decoding bootstraps the development of orthographic representations (Share, 1995; Ziegler, Perry, & Zorzi, 2014). Finally, sequential generation of letters is a prominent feature of written spelling, and it is a key aspect in popular computational models of spelling (Glasspool & Houghton, 2005; Houghton, Glasspool, & Shallice, 1995).

To learn the sequential structure of words, the network was exposed to a corpus of English monosyllables, with each word presented one letter at a time. After learning, the generative model was expected to have inferred the orthographic structure underlying the training data. To assess this, we first evaluated the accuracy of context-dependent predictions (Section 3, illustrated in Fig. 2A). Moreover, the model should be able to reproduce the training sequences and generalize, thus producing well-formed pseudowords (Section 4).

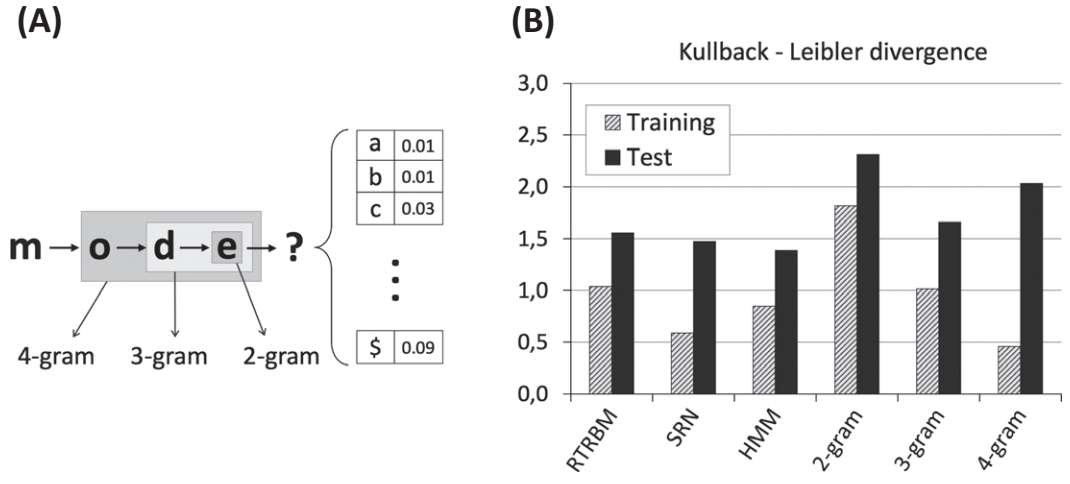

Fig. 2. (Panel A) A prototypical prediction problem, in which a certain context is given (i.e., the first four letters of a word) and the aim is to predict the probability distribution of the following letter. (Panel B) Prediction error on training set (gray) and test set (black) for different models, measured as the KL-divergence between predicted and empirical distributions (small is better).

3. Learning graphotactics

3.1. Dataset

We used a dataset of 6,670 English monosyllables of variable length (from three to seven letters) extracted from CELEX (Baayen, Piepenbrock, & van Rijn, 1993), which is an electronic corpus that comprises lexicons of British English, German, and Dutch. The dataset was randomly split into a training set of 5,300 words and a test set of 1,370 words. Words were codified as sequences of letters, represented with fixed-size binary orthogonal vectors of 27 units (one for each possible letter, plus a termination symbol).

3.2. RTRBM training

The RTRBM model was trained using the CD learning algorithm. Following Sutskever et al. (2008), CD started with few iterations (CD-5) and gradually increased as learning proceeded (until CD-40). Learning rate, number of hidden units and number of learning epochs were tuned with the aim of obtaining high prediction accuracy on the training data. In particular, parameter tuning resulted in a learning rate of 0.1, 200 hidden units, and learning was stopped when no significant improvements occurred on the training set (after approximately 300 epochs). We did not tune the parameters to maximize generalization performance (e.g., using cross-validation) because this procedure is implausible from a cognitive modeling perspective. Training times were significantly reduced by exploiting multi-core graphical processors (Testolin et al., 2013; Tieleman, 2010) and by

adopting a mini-batch learning scheme (mini-batch size = 50), obtaining a speed-up of about 25 times relative to a quad-core CPU implementation. The complete source code of our model is publicly available for download.[1]

3.3. Alternative models: SRNs, n-grams, HMMs

We compared the prediction performance of the network with that of SRNs (Elman, 1990) and to that of other two popular families of probabilistic generative models for sequential data, *n*-grams (Brown, DeSouza, Mercer, Della Pietra, & Lai, 1992) and HMMs (Rabiner, 1989).

Simple recurrent networks are feed-forward neural networks composed by three layers. The input layer contains both the current element of the sequence that is being processed and contextual information encoded by the network, which is simply a copy of the hidden layer activity at the previous time step. At the beginning of a sequence, the activations of context units are usually set to zero. An output layer is used to perform a prediction of the next element of the sequence, and learning is performed by back-propagating the mismatch error between the network prediction and the desired target value. At each prediction step the vector of output unit activations was normalized (i.e., made to sum up to 1) to obtain conditional probability distributions. This type of output is more appropriate for the comparison between SRNs and probabilistic models, and it is also a prerequisite for extending the classic SRN into a stochastic version (SSRN, see below) that can be used to spontaneously generate sequences. As for the RTRBM, the learning parameters of the SRN were tuned to maximize accuracy on the training data. The resulting network had 200 hidden neurons; thereby the two connectionist models had about the same number of connections (the SRN had slightly more connections due to the additional set of hidden-to-output weights, as also shown in Fig. 1D). This implies that both models had approximately the same complexity (i.e., the same number of parameters to be fit). Learning rate was set to 0.01 and training was performed for 250 epochs.

The *n*-gram models were implemented as look-up tables, where each row contained the successor distribution computed from the training data for each possible context (i.e., the $n - 1$ preceding letters), with n varying between 2 and 4 (see Fig. 2A). These models therefore treat two sequences as equivalent if they end in the same $n - 1$ letters: assuming a value $k \geq n$, it holds that

$$P(l_k|l_1^{k-1}) = P(l_k|l_{k-n+1}^{k-1}),$$

where l_1^{k-1}represents the sequence of letters $l_1 l_2 \ldots l_{k-1}$ and l_k is the k-th letter of a word. Although this might seem a somewhat crude approximation, *n*-grams have demonstrated very good performances (Brown et al., 1992) and still constitute a reference framework for language modeling. One of the major drawbacks of *n*-grams is caused by data sparsity: Items not present in the training set will be given a probability of zero, which

motivates the use of smoothing techniques. In our study, we used a simple form of additive smoothing (Chen & Goodman, 1996).

We also tested HMMs of first- and second-order, with a number of states ranging from 7 to 60 following a previous study (Sang & Nerbonne, 1999). HMMs assume that the system being studied can be modeled as a Markov process with a certain number of unobserved (i.e., hidden) states. In first-order models, the probability of being in a certain state at the current time step only depends on the state of the model at the immediately preceding time step. In second-order models, this dependence is extended to the last two states. Each state has an associated emission distribution that describes the probability of emitting (i.e., observing) each symbol of the alphabet from that state. A transition distribution specifies the probability of moving from each hidden state to any other. If two states are not connected, the corresponding transition probability will be zero. Finally, an initial state distribution specifies the probability of starting the generation of a sequence from each of the states of the model. The parameters of an HMM can be estimated using an iterative procedure known as the Baum–Welch algorithm (Rabiner, 1989), which adjusts the probability distributions to raise the likelihood of the training data using an expectation-maximization method. As for the other models, HMM hyper-parameters were tuned to maximize accuracy on the training data. In particular, the highest performance was obtained using a first-order model with 40 hidden states, trained for 10,000 iterations with a likelihood cut-off of 0.001 and 1,000 steps in the Baum–Welch algorithm.

3.4. Evaluation of context-dependent predictions

We evaluated the performance of all models on predicting the next element of a sequence, given a certain context. Accuracy was measured as mean prediction error on both training and test sets using a computationally efficient procedure that exploits a tree-based data structure (Stoianov, 1998). In particular, we evaluated the response of each model across all possible left contexts in the evaluation sets (i.e., variable length, initial parts of words). To this aim, we created a k-tree data structure, where k is the size of the alphabet (26 letters plus one termination symbol). Words were encoded as paths in the tree, starting from the root. Every node in the tree represents a left context (which is the path from the root to the current node) and it might have a number of children or alternatively constitute the end of a word. It is possible to efficiently compute the *empirical successor distribution* of each context in the dataset by counting the frequency of each child of a node (i.e., the frequency of each letter following that context) and normalizing the resulting vector to sum up to 1. Once the empirical successor distribution has been computed for all the variable length contexts in the dataset, each model is probed with all possible contexts to compute the *predicted successor distributions*. The vectors of empirical and predicted successor distributions can then be compared according to some metric to measure the discrepancy between observed and predicted values. We used the Kullback–Leibler (KL) divergence as distance metric (Kullback & Leibler, 1951), which measures the difference between the two probability distributions as:

$$D_{KL}(E||M) = \sum_{i=1}^{k} \log \frac{E_i}{M_i} E_i,$$

where E is the empirical distribution, M is the predicted (model) distribution, and k is the size of the alphabet. $D_{KL}(E||M)$ measures the information lost when M is used to approximate E and it tends to approach zero as the two distributions become more similar (i.e., when the model makes accurate predictions). We preferred the KL-divergence over other metrics (e.g., Euclidean distance or cosine similarity) because of its sound probabilistic interpretation and its direct link to the notions of cross-entropy and perplexity, which are two other metrics commonly used to assess language models. Nevertheless, it is worth noting that the results reported below are robust with respect to the type of metric (see Testolin, Sperduti, Stoianov, & Zorzi, 2012, for a preliminary study based on the Euclidean distance measure). The total error of each model was the average KL-divergence across all possible contexts in the evaluation datasets.

To compute the predicted successor distribution for the RTRBM, a response was collected by sequentially clamping the visible units on the given letters (i.e., left context) and letting the network generate visible layer activations. The normalized activations (i.e., summing up to 1) constitute the predicted successor distribution M, which corresponds to the conditional probability distribution of all letters in the alphabet given the context $v_1, v_2, \ldots, v_{t-1}$ encoded by the hidden unit activation h_{t-1}:

$$M \leftarrow P(v_t|v_1, v_2, \ldots, v_{t-1}) \approx P(v_t|h_{t-1}).$$

A vector M representing the predicted successor distribution for each context was also obtained for the other models tested. For SRNs, we collected the output values (normalized to sum up to 1) in response to a given context (i.e., the sequence of preceding letters). For n-grams, the successor distribution directly corresponded to the row associated with a particular context. For HMMs, the optimal sequence of hidden states (i.e., the one with the highest probability under the current context) was first computed using the Viterbi algorithm, and then the successor distribution was read out from the emission probabilities of the last state of the sequence.

3.5. Results

Prediction errors for the different models are plotted in Fig. 2B. The prediction accuracy of the RTRBM on the test set (black columns) was higher than that of all n-gram models, and just slightly lower than that of SRN and HMM. It can be noted that the 2-gram model is inadequate for accurately predicting which letter will follow a given context, because it only takes into account the last letter as context. Other models reach better accuracy thanks to their ability to consider longer context when making predictions. Moreover, the results confirmed a critical limitation of n-gram models, which is their poor generalization (Brown et al., 1992). The longer the context of the n-gram, the

greater was the prediction accuracy on the training data. However, on test data the performance of the 4-gram model significantly decreased due to coding of too specific contexts. This limitation could be alleviated by using more sophisticated smoothing techniques. On the other hand, both connectionist models avoid the problem of specificity by exploiting distributed representations of the context, which turns into good generalization performance.

Overall, the results show that both the RTRBM and the SRN successfully learned the context-dependent transition probabilities between the letters of English words, yielding a level of accuracy that is comparable to that of other popular sequential language models.

4. Generative abilities

Having assessed the prediction accuracy of the different models, we then investigated their ability to autonomously generate well-formed sequences of letters. Indeed, a generative model will produce a sequence of letters even when there is no external context to drive the generation. Due to the deterministic, input-driven nature of the SRN, its basic version cannot be used to autonomously generate sequences. We therefore propose a simple extension of the model that allows it to produce sequences from the learned probability distribution. We evaluated the quality of the letter sequences generated by the learning models by comparing them with those contained in the training and test sets, and with those produced by two published pseudoword generator algorithms.

4.1. Augmenting the SRN: Stochastic sampling for sequence generation

Stochastic sampling was implemented by first transforming the SRN's output activations into a (conditional) probability distribution of letters. As the coding scheme is orthogonal, whereby each letter is coded by a specific unit, the probability distribution could be obtained by simply normalizing the vector of output units to sum up to 1. The next element of the generated sequence can then be selected by using an external stochastic process that samples one letter according to the conditional probabilities. A straightforward way to realize this is to calculate the corresponding cumulative distribution, and then select the letter corresponding to a random number drawn from the interval [0, 1]. This procedure does not require any selection threshold, because all letters with non-zero probability can potentially be chosen to be the successor. The selected element is given as input to the network at the next time step, and this process is repeated until the termination symbol is produced. A similar approach has been recently applied to a different class of recurrent neural networks (Sutskever et al., 2011). Even if we used logistic sampling, it should be noted that other approaches can be used to obtain a probability distribution in the output units, for example, by using a softmax function (see McClelland, 2013; for a discussion about the relation of logistic and softmax sampling and their Bayesian interpretation). A graphical representation of the stochastic SRN (SSRN) is

provided in Fig. 1C, whereas Fig. 1D illustrates the generation process of a three-element sequence when the network is unfolded in time.

4.2. Pseudoword generators

Pseudoword generators are commonly used in psycholinguistic research. They provide a useful benchmark for assessing the quality of letter sequences because they have been engineered to maximize the well-formedness of the generated items. We considered the ARC pseudowords database (Rastle, Harrington, & Coltheart, 2002) and Wuggy (Keuleers & Brysbaert, 2010). The ARC database contains 310,000 non-pseudohomophonic monosyllabic pseudowords, built using a hand-crafted grammar that defines phonological constraints on monosyllables. A set of phoneme-to-grapheme correspondences extracted from CELEX is used to derive possible spellings of legal phonological strings, which are then converted back to phonological representations using a set of grapheme–phoneme correspondences. Finally, phonological strings that differ from the initial phonologies are excluded from the database. The Wuggy pseudoword generator takes a different approach. Instead of combining subsyllabic elements like in the ARC database, it starts from a given set of words, which are syllabified and used to build a bigram chain. Pseudowords are then generated by recursively iterating through the chain (Keuleers & Brysbaert, 2010). Wuggy is particularly interesting for our comparison, because it does not use phonological representations and it starts the generation from a reference list of words. Thus, we could generate pseudowords using the same training set of our RTRBM. It is worth stressing that the pseudoword generators should not be considered as a "gold-standard" to assess the performance of the models, because they do not necessarily represent human performance. Nevertheless, the reference to ARC and Wuggy provides a useful benchmark for assessing the performance of learning models with respect to carefully engineered algorithms.

4.3. Evaluation of generative performance

Each model was used to generate an arbitrary number of sequences, which was chosen to be 300 times the size of the training set (i.e. 1,590,000 samples). We then calculated two indexes:

- *Completeness* of the generation, computed as the ratio between the number of sampled sequences that also appeared in the training set (without repetitions) and the total number of sequences contained in the training set;
- *Fidelity* of the generation, computed as the ratio between the sampled sequences that also appeared in the training set (possibly repeated) and the total number of sampled sequences.

The first indicator describes the ability of the model to regenerate the training sequences and it depends on the sampling size. Augmenting the number of the samples generally increases the completeness of the generation. The second indicator does not

depend on the size of the sampling and gives an idea about the model tendency to generate new wordforms instead of reproducing only previously seen sequences: the lower the fidelity, the greater this tendency.

Given that all models generated a consistent amount of new wordforms (as reported below), we inspected the quality of the generated strings that did not belong to the training set. All models produced a number of real English words that were not part of the training set, which we excluded from the analysis to allow a fair comparison with the pseudoword generator algorithms. We therefore analyzed the 20,000 most frequently generated pseudowords composed by at least three letters. We randomly selected the same number of pseudowords (with at least three letters) from the ARC database for a comparison with its underlying generation algorithm. The Wuggy pseudoword generator was supplied with the words of the training dataset as input to build the bigram chain, and we selected the 20,000 most frequently generated pseudowords using the following parameter set: maximal number of candidates was set to 15, maximal search time was set to 10 s, and all output restrictions were required (i.e., match length of subsyllabic segments, match letter length, match transition frequencies and match subsyllabic segments). The set of pseudowords generated by each model or algorithm was evaluated in terms of the following statistical features (Duyck, Desmet, Verbeke, & Brysbaert, 2004):

- *Sequence length*, that we expected to be close to the average length of words in the training set;
- *Orthographic neighborhood* (*Coltheart's N*): The number of orthographic neighbors that a string has. An orthographic neighbor is a word of the same length that differs from the original string by only one letter. For example, given the pseudoword "cat," the words "bat," "fat," and "cab" are orthographic neighbors;
- *Constrained bigrams and trigrams frequency*: Averaged type frequency of constrained bigrams (trigrams) for the wordform. A constrained bigram is defined as a specific two-letter combination in a specific position and specific word length. That is, "*es*" in "b*es*t" is considered the same as in "n*es*t," but is different both in "y*es*" (different length) and in "do*es*" (different position).

4.4. Results

The results of the generation process in terms of completeness and fidelity are shown Table 1.

With regard to the RTRBM and the SSRN, Fig. 3 shows that both indicators improved as training proceeded and that eventually both networks were able to generate a large fraction

Table 1
Completeness and fidelity of the generation process for each tested model

	RTRBM (%)	SSRN (%)	HMM (%)	2-gram (%)	3-gram (%)	4-gram (%)
Completeness	91	98	92	67	96	99
Fidelity	16	37	13	5	21	58

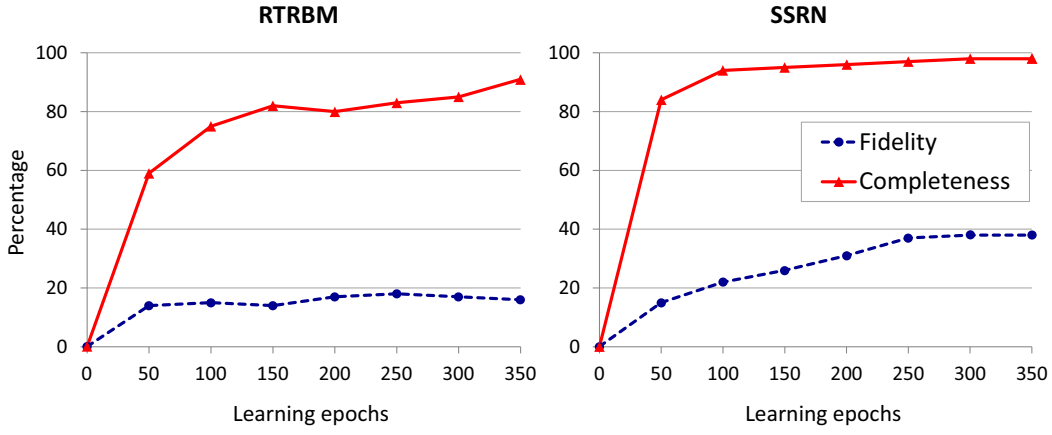

Fig. 3. Recurrent Temporal Restricted Boltzmann Machine (RTRBM) and SSRN completeness and fidelity (percentage) of generation as a function of training time (in epochs).

of the words in the training set. Nevertheless, the low fidelity suggests that both models are not encoding entire sequences, but they rather exploit local transition rules during the generative process. Indeed, the networks generated many legal sequences that were not present in the training set (a few samples generated by the RTRBM are reported in Fig. 4A). Some of these sequences were in fact real English words that were not part of the training set, and a similar pattern was found for the other models (see first row of Table 2 for details). It is worth noting that the generated pseudowords might be composed by "legal" bigrams (i.e., combinations of two subsequent letters that are observed in the training set) or by novel (and potentially illegal) bigrams. To investigate this point, we computed the fraction of illegal bigrams produced by the RTRBM with respect to the total generated bigrams in the considered set of pseudowords. We found a ratio of 0.2%, which confirms that the RTRBM generated many novel pseudowords without introducing illegal bigrams.

Results of the analysis of pseudoword quality for all models are reported in Table 2. The number of pseudowords shorter than three letters that was excluded from the analysis was very small for all the models. As expected, the average length of the pseudowords generated by the different models was similar to that of the words in the training set, except for the 4-gram model, which produced longer sequences. Interestingly, only the RTRBM and Wuggy never generated words longer than seven letters, which is the maximum length of the words in the training data (note that Wuggy was explicitly required to respect this constraint). All the other models, instead, generated a certain number of pseudowords longer than seven letters. This ranged from just 16 for the SSRN to several thousand for the 4-gram model.

Importantly, pseudowords generated by connectionist models had the highest mean orthographic neighborhood (4.96 for the RTRBM and 5.03 for the SSRN), followed by those generated by the 3-gram model and Wuggy (Fig. 4C). On the other hand, ARC pseudowords had the lowest orthographic neighborhood (0.56). The mean constrained bigram frequency (Fig. 4D) was higher for the RTRBM compared to all other models,

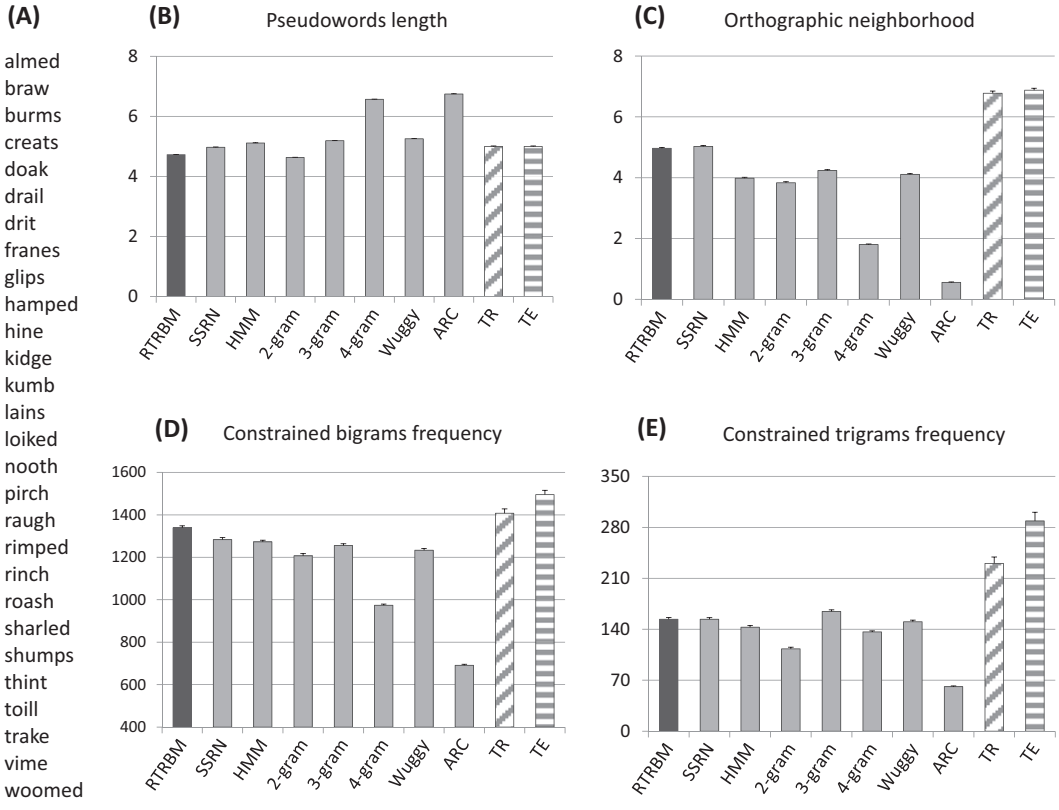

Fig. 4. (Panel A) Sample of pseudowords generated by the Recurrent Temporal Restricted Boltzmann Machine (RTRBM). Average length (Panel B), average orthographic neighborhood (Panel C), constrained bigrams frequency (Panel D), and constrained trigrams frequency (Panel E) collected over 20,000 pseudowords generated by different models and over words of the training (TR) and test (TE) datasets.

approaching the value of the words in the training set. Both these measures indicate that the RTRBM produced high-quality pseudowords. Note that the statistics computed over the training set are very close to those computed over the test set (compare the last two columns of Table 2), thereby showing that these values are representative of the statistical distribution in English monosyllabic words.

It is also interesting to note that the RTRBM, the SSRN, and Wuggy generated pseudowords with similar statistics, even if the latter exploits a sophisticated algorithm based on bigram chains that are carefully constructed taking into account linguistic information and that are processed using an optimized search procedure (Keuleers & Brysbaert, 2010). At the same time, trigram frequencies (Fig. 4E) computed on the pseudowords generated by all the tested models are substantially lower than those computed on the words contained in the training and test sets. This result indicates a partial mismatch between existing models and their ability to encode actual distal statistics that are present in the word corpus.

5. Discussion

In this study, we modeled the process of learning orthographic structure from sequences of letters using a recently proposed generative neural network, the Recurrent Temporal Restricted Boltzmann Machine (RTRBM; Sutskever et al., 2008). We showed that this sequential network is able to learn the structure of English monosyllables in a completely unsupervised fashion, by only trying to accurately reproduce input wordforms presented one letter at a time.

We first demonstrated that the RTRBM successfully learned graphotactics by testing its performance on a prediction task, where initial parts of words were given as a context and the network predicted the probability distribution of the following letters. The RTRBM yielded a prediction performance comparable to that of a simple recurrent network (SRN), which is the most widely used connectionist architecture to model sequential data. We also compared the RTRBM with other popular (non-connectionist) probabilistic generative models, Hidden Markov Models and *n*-grams, which constitute the state-of-the-art in several sequence learning tasks but do not provide insights in terms of the underlying neural computations. We then assessed the generative ability of the considered models by letting them to autonomously produce sequences of letters and measuring their well-formedness. In particular, we calculated the mean length, orthographic neighborhood and constrained bigram and trigram frequencies of the generated sequences, and we used these indicators to compare the quality of these pseudowords with that of the words in the training and test sets, and with that of two popular pseudoword generators used in psycholinguistic studies, namely the ARC non-words database (Rastle et al., 2002) and Wuggy (Keuleers & Brysbaert, 2010). We found that the RTRBM produced very high-quality pseudowords, thus confirming that it correctly learned the orthographic structure of English monosyllables. In this regard, it is worth noting that the results of our model should readily extend to other alphabetic languages. It should also be emphasized that our assessment of the generative ability of the network was not performed over a sample of pseudowords generated by human subjects. This constitutes an important future research direction because, to our knowledge, there is no study that has systematically investigated the spontaneous production of pseudowords by humans. An empirical study on human pseudoword generation would provide a critical baseline to measure the quality of different models.

To allow autonomous generation of sequences with the SRN, which is inherently deterministic and input-driven (i.e., bottom–up), we extended its basic formulation. We therefore implemented a stochastic variant of the SRN, the SSRN, in which the activations of the output units are first normalized to be treated as conditional probability distributions. An external stochastic process is then used to sample the next element of the sequence, which is fed back as input to the network at the following time step. Interestingly, the pattern of results obtained with the RTRBM was very similar to that obtained with the SSRNs. On the one hand, this is not surprising because both connectionist models try to predict the next element of a sequence by learning conditional probabilities from the training data. Indeed, there is a tight formal relationship between probabilistic graphical

Table 2
Statistics on existing English words generated, pseudoword length (average, maximum, shorter than three and longer than eight), orthographic neighborhood (OrthN), constrained bigram frequency (BigFreq), and constrained trigram frequency (TriFreq) for all the tested models and pseudoword generators

	RTRBM	SSRN	HMM	2-gram	3-gram	4-gram	Wuggy	ARC	TR	TE
Existing	1,461	1,869	1,228	1,060	1,665	1,550	0	0	5,300	1,370
Length	4.73 ± 0.01	4.97 ± 0.01	5.12 ± 0.01	4.63 ± 0.01	5.19 ± 0.01	6.57 ± 0.01	5.26 ± 0.01	6.75 ± 0.01	5.00 ± 0.02	5.00 ± 0.03
MaxLength	7	8	12	9	8	12	7	12	7	7
Length < 3	81	24	98	329	90	2	0	0	0	0
Length > 8	0	16	201	35	81	4,416	0	5,637	0	0
OrthN	4.96 ± 0.03	5.03 ± 0.03	3.98 ± 0.03	3.84 ± 0.03	4.23 ± 0.03	1.80 ± 0.02	4.11 ± 0.03	0.56 ± 0.01	6.78 ± 0.07	6.88 ± 0.14
BigFreq	1,339 ± 9.79	1,283 ± 9.54	1,273 ± 7.95	1,207 ± 10.88	1,255 ± 8.42	974 ± 5.88	1,233 ± 8.38	691 ± 4.85	1,408 ± 20.44	1,495 ± 56.81
TriFreq	154 ± 2.51	154 ± 2.24	143 ± 2.25	113 ± 2.27	165 ± 2.15	136 ± 1.47	150 ± 2.17	61 ± 0.89	230 ± 8.84	289 ± 47.02

Note. The last two columns report the same measures for the complete words of the training (TR) and test (TE) datasets (± indicates the standard error for each measure).

models and recursive neural networks (Baldi & Rosen-Zvi, 2005). However, the two architectures also differ in several fundamental aspects. The SSRN learns a mapping function between the current input (plus temporal context) and a separate output representation. In contrast, the RTRBM learns an internal model of the data (i.e., the hidden causes that generated the input patterns) by trying to accurately reproduce the incoming information through feedback (i.e., top–down) connections. That is, sequential information is learned by trying to re-generate the training sequences on the same layer that is used for providing the input. Moreover, the SSRN relies on two additional operations, one that performs a non-local normalization over the activations of output units and another that samples the predicted element exploiting an external, *ad-hoc* stochastic process. In contrast, autonomous sequence generation from the RTRBM is an intrinsic feature of the network: There is no need to perform normalization and to sample from the corresponding distribution because the probabilistic behavior is caused by the stochastic dynamics that is also a crucial part of the learning process. Nevertheless, also the SSRN might be appealing as a cognitive modeling architecture due to its much simpler formulation and its close relationship to the widely used SRNs. On the other hand, the RTRBM is the only choice when the learning task involves multimodal, distributed representations as input to the network (Sutskever et al., 2008) instead of the simpler, localistic scheme adopted in our simulations. Indeed, in such a case it is not possible to choose which element should be generated at the next time step using the straightforward sampling scheme implemented in the SSRN.

In a broader perspective, stochastic generative neural networks represent an appealing framework for modeling cognitive phenomena because they have a sound probabilistic formulation and incorporate key principles of neural computation that are not captured by classic connectionist architectures (see Zorzi et al., 2013; for discussion). For example, bidirectional activation propagation (interactivity) seems to be a fundamental characteristic of information flow in the cortex (McClelland, Mirman, Bolger, & Khaitan, 2014; O'Reilly, 1998) and the model-driven, top–down generation of activation over sensory layers is consistent with a Bayesian view of perception as well as with a predictive coding framework (Clark, 2013; Huang & Rao, 2011). Moreover, stochastic generative networks like Boltzmann Machines only rely on locally available signals to efficiently compute the global state of the system. However, these networks did not become popular among connectionist modelers due to the very high computational demands of their original learning algorithm. Recent advances in the theory of graphical models (Jordan & Sejnowski, 2001; Koller & Friedman, 2009), the introduction of an efficient learning procedure (Hinton, 2002), and the extension to hierarchical architectures (Hinton & Salakhutdinov, 2006) have paved the way to the deep learning framework, which represents a major breakthrough for the connectionist modeling enterprise (for reviews see Hinton, 2013; Zorzi et al., 2013). Although generative neural networks are being successfully applied in cognitive (neuro)science modeling, their use so far has been primarily focused on the investigation of perceptual tasks with static information (e.g., space coding, De Filippo, De Grazia, Cutini, Lisi, & Zorzi, 2012; numerosity estimation, Stoianov & Zorzi, 2012; visual word recognition, Di Bono & Zorzi, 2013). The RTRBM extends

the application of generative networks to the temporal domain, where the system has also to extract the dynamic aspect of sequentially presented input. In this regard, it is worth noting that the RTRBM is not a deep network, but it could be used as a building block of a deep architecture for learning a hierarchical generative model of the sequential data.

Language is a prominent example of a domain where sequential processing is ubiquitous, data dynamics is highly structured, and information is integrated over time. For this reason, the problem of discovering structure in time is a key issue for connectionist modeling of language (Elman, 1990). Even if there is clear evidence that statistical learning principles underlie several basic linguistic abilities (Romberg & Saffran, 2010), it has been argued that the great complexity of language (and, more generally, of knowledge representation) might be better captured by probabilistic models defined over rich symbolic structures, in which learning and processing are seen as problems of induction and inference (Chater & Manning, 2006; Chater, Tenenbaum, & Yuille, 2006; Griffiths et al., 2010). Though the probabilistic formulation greatly improves the descriptive capability of symbolic rules and representations, the appeal of connectionism is that it shows how structure can emerge as high-order statistical features of the input from learning within a neurally plausible architecture (McClelland et al., 2010). Generative neural networks can be formally defined within the powerful framework of probabilistic graphical models (Jordan & Sejnowski, 2001), which also represents the basis for a broad class of structured Bayesian models (Koller & Friedman, 2009). Probabilistic graphical models provide a general approach to model complex statistical distributions involving a large number of stochastic variables that interact together. Basically, the topology of a graph defines the scope of interaction and the conditional dependencies between random variables, thereby allowing to compactly represent joint distributions through factorization. This permits to derive efficient inference and learning procedures, which can often be implemented through operations that are local with respect to the structure of the graph (Pearl, 1988). We therefore argue that generative neural networks constitute a promising avenue for research in computational cognitive science, because they can bridge the gap between emergentist connectionist approaches and structured Bayesian models of cognition (Zorzi et al., 2013). Some attempts to integrate these two approaches have recently led to a compositional architecture that learns a hierarchical Dirichlet process prior over the activities of the top-level features in a deep Boltzmann Machine, allowing to learn novel concepts from very few examples (Salakhutdinov, Tenenbaum, & Torralba, 2011).

Sequential statistical learning is a general phenomenon that is found across sensory modalities (Conway & Christiansen, 2005). Although it is not yet clear whether the underlying mechanism is unitary or modality constrained, it is interesting to note that the sequential generative network used in our study has been previously applied to modeling video sequences (Sutskever et al., 2008). The RTRBM was able to capture the high-dimensional, multimodal nature of the pixels distribution across subsequent frames, despite the complex nonlinearities characterizing the movies of the dataset. An extension of the RTRBM, called RNN-RBM, has also recently been used to model temporal dependencies in polyphonic music (Boulanger-Lewandowski, Bengio, & Vincent, 2012), thereby supporting the intriguing hypothesis that music, language, and statistical learning

might be tightly linked (McMullen & Saffran, 2004; Patel, 2003). In our work, we did not consider the RNN-RBM as a reference architecture due to its increased complexity, which is caused by the fact that the network is composed by two separate modules, one recurrent neural network (RNN) that propagates the contextual information over time and one RTRBM that models the conditional distributions at each time step. It is not clear whether this separation can be justified from a psychological perspective, but it would certainly be an interesting research direction to also investigate the potential of the RNN-RBM to model sequential cognitive tasks. Moreover, it has recently been shown that even a simple RNN can significantly outperform the more complex RNN-RBM on music prediction if weights are properly initialized through an effective pre-training (Pasa & Sperduti, 2014; Pasa, Testolin, & Sperduti, 2014). It would therefore be interesting to explore in which way such pre-training schemes could also affect learning of orthographic structure.

While our results on learning orthographic structure are encouraging for the application of RTRBMs to language learning problems, there are several important questions that should be addressed in future psycholinguistic modeling research. First, analysis of the hidden units' dynamics might reveal the nature of the temporal features extracted by the RTRBM, for example, some neurons might become sensible to particular transition rules (e.g., see Nerbonne & Stoianov, 2004; Stoianov, 2001). Second, a systematic investigation of the "memory span" of the network would allow estimating how much contextual information can be maintained in the hidden layer. If the network is able to encode entire sequences at the hidden layer in a way that allows discrimination of different words, such static distributed representation of sequences could be used as input to higher level networks/modules (Sibley et al., 2008; Stoianov, 1999). This would allow to better investigate the extent to which unsupervised statistical learning could generate novel word-like units (Saffran, 2001), thereby showing how syntactic structures could emerge by first segmenting the words from continuous speech, and subsequently discovering the permissible orderings of the words (Saffran & Wilson, 2003). As noted before, the shallow architecture of the RTRBM could serve as the basis for a deeper generative model, which is likely to increase its ability to discover complex structure hidden in sequential data and expand its applicability to a broader range of phenomena. In this respect, it is worth mentioning that there exist some other recent models, like the Conditional RBM (Taylor, Hinton, & Roweis, 2011), which might be more easily stacked into a hierarchical system. Finally, the application of the model should also be investigated on linguistic tasks, such as word segmentation and decomposition, in which probabilistic generative models represent the state-of-the-art (Creutz & Lagus, 2002; Spiegler, Golenia, & Flach, 2010). The RTRBM could also represent a very useful platform for studies of language similarities and dialectology (Nerbonne & Heeringa, 2009).

In conclusion, our study suggests that sequential, stochastic generative networks are appealing for modeling dynamic cognitive processes. We believe that this line of research holds great promise to build more powerful connectionist architectures for representing and manipulating structured information, using hierarchies of processing levels that operate across different temporal and spatial scales.

Acknowledgments

This study was supported by the European Research Council (grant no. 210922 to M.Z.). I. S. was supported by a Marie Curie Intra European Fellowship PIEF-GA-2013-622882 within the 7th Framework Programme.

Note

1. http://ccnl.psy.unipd.it/research/RTRBM.

References

Abbott, L. F. (2008). Theoretical neuroscience rising. *Neuron*, *60*(3), 489–495. doi: 10.1016/j.neuron.2008.10.019.

Ackley, D., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*(1), 147–169. doi: 10.1016/S0364-0213(85)80012-4.

Baayen, R. H., Piepenbrock, R., & van Rijn, H. (1993). *The CELEX lexical database [CD-ROM]*. Philadelphia: Linguistic Data Consortium, University of Pennsylvania.

Baldi, P., & Rosen-Zvi, M. (2005). On the relationship between deterministic and probabilistic directed Graphical models: From Bayesian networks to recursive neural networks. *Neural Networks*, *18*(8), 1080–1086. doi: 10.1016/j.neunet.2005.07.007.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127. doi: 10.1561/2200000006.

Bengio, Y., Courville, A., & Vincent, P. (2012). Representation learning: A review and new perspectives. arXiv, 1206.5538, 1–34.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*, 157–166.

Berkes, P., Orbán, G., Lengyel, M., & Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, *331*(6013), 83–87. doi: 10.1126/science.1195870.

Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In J. Langford & J. Pineau (Eds.), *International Conference on Machine Learning* (pp. 1159–1166). Madison, WI: Omnipress.

Brown, P., DeSouza, P., Mercer, R., Della Pietra, V., & Lai, J. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–479.

Chater, N., & Manning, C. D. (2006). Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences*, *10*(7), 335–344. doi: 10.1016/j.tics.2006.05.006.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*(7), 287–291. doi: 10.1016/j.tics.2006.05.007.

Chen, S., & Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In A. Joshi & M. Palmer (Eds.) *Proceedings of the 34th annual meeting of the association for computational linguistics* (pp. 310–318). Stroudsburg, PA: Association for Computational Linguistics.

Chomsky, C. (1970). Reading, writing, and phonology. *Harvard Educational Review*, *40*(2), 287–309.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *The Behavioral and Brain Sciences*, *36*(3), 181–204. doi: 10.1017/S0140525X12000477.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology. General*, *120*(3), 235–253.

Conway, C. M., & Christiansen, M. H. (2005). Modality-constrained statistical learning of tactile, visual, and auditory sequences. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *31*(1), 24–39. doi: 10.1037/0278-7393.31.1.24.

Creutz, M., & Lagus, K. (2002). Unsupervised discovery of morphemes. *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, *6*, 21–30.

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz machine. *Neural Computation*, *7*(5), 889–904.

De Filippo De Grazia, M., Cutini, S., Lisi, M., & Zorzi, M. (2012). Space coding for sensorimotor transformations can emerge through unsupervised learning. *Cognitive Processing*, *13*(Suppl 1), 141–146. doi: 10.1007/s10339-012-0478-4

Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., & Ng, A. Y. (2012). Large scale distributed deep networks. *Advances in Neural Information Processing Systems*, *24*, 1–9.

Di Bono, M. G., & Zorzi, M. (2013). Deep generative learning of location-invariant visual word recognition. *Frontiers in Psychology*, *4*(September), 635. doi: 10.3389/fpsyg.2013.00635.

Duyck, W., Desmet, T., Verbeke, L. P. C., & Brysbaert, M. (2004). WordGen: A tool for word selection and nonword generation in Dutch, English, German, and French. *Behavior Research Methods, Instruments, & Computers*, *36*(3), 488–499.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Elman, J. L., Bates, E., Johnson, M., Karmiloff-smith, A., Parisi, D., & Plunkett, K. (1996). New perspectives on development. In J. L. Elman (Ed.) *Rethinking innateness: A connectionist perspective on development* (pp. 1–45). Cambridge, MA: MIT Press.

Fiser, J., & Aslin, R. N. (2001). Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological Science*, *12*(6), 499–504. doi: 10.1111/1467-9280.00392.

Friston, K. J. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *360*(1456), 815–836. doi: 10.1098/rstb.2005.1622.

Glasspool, D. W., & Houghton, G. (2005). Serial order and consonant-vowel structure in a graphemic output buffer model. *Brain and Language*, *94*(3), 304–330. doi: 10.1016/j.bandl.2005.01.006.

Glasspool, D. W., Shallice, T., & Cipolotti, L. (2006). Towards a unified process model for graphemic buffer disorder and deep dysgraphia. *Cognitive Neuropsychology*, *23*(3), 479–512. doi: 10.1080/02643290500265109.

Grainger, J., Dufau, S., Montant, M., Ziegler, J. C., & Fagot, J. (2012). Orthographic processing in baboons (Papio papio). *Science*, *336*(6078), 245–248. doi: 10.1126/science.1218152.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences*, *14*(8), 357–364. doi: 10.1016/j.tics.2010.05.004.

Hauser, M. D., Newport, E. L., & Aslin, R. N. (2001). Segmentation of the speech stream in a non-human primate: Statistical learning in cotton-top tamarins. *Cognition*, *78*(3), B53–B64.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*(8), 1771–1800.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in Cognitive Sciences*, *11*(10), 428–434.

Hinton, G. E. (2013). Where do features come from? *Cognitive Science*, *1–24*, doi:10.1111/cogs.12049.

Hinton, G. E., & Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *352*(1358), 1177–1190.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, volume 1: Foundations* (pp. 77–109). Cambridge, MA: MIT Press.

Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. doi: 10.1126/science.1127647.

Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, *98*(1), 74–95.

Houghton, G., Glasspool, D. W., & Shallice, T. (1995). Spelling and serial recall: Insights from a competitive queuing model. In G. D. A. Brown & N. C. Ellis (Eds.), *Handbook of spelling: Theory, process and intervention* (pp. 365–404). Chichester, England: John Wiley.

Huang, Y., & Rao, R. P. N. (2011). Predictive coding. *Wiley Interdisciplinary Reviews. Cognitive Science*, *2* (5), 580–593. doi: 10.1002/wcs.142.

Jain, A. K., Duin, P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37. doi: 10.1109/34.824819.

Jordan, M. I., & Sejnowski, T. J. (Eds.) (2001). *Graphical models: Foundations of neural computation*. Cambridge, MA: MIT Press.

Keuleers, E., & Brysbaert, M. (2010). Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, *42*(3), 627–633. doi: 10.3758/BRM.42.3.627.

Kok, P., Jehee, J. F. M., & de Lange, F. P. (2012). Less is more: Expectation sharpens representations in the primary visual cortex. *Neuron*, *75*(2), 265–270. doi: 10.1016/j.neuron.2012.04.034.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. Cambridge, MA: The MIT Press.

Krogh, L., Vlach, H. A., & Johnson, S. P. (2013). Statistical learning across development: Flexible yet constrained. *Frontiers in Psychology*, *3*, 598. doi: 10.3389/fpsyg.2012.00598

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Marcus, G. F. (1999). Rule learning by seven-month-old infants. *Science*, *283*(5398), 77–80. doi: 10.1126/science.283.5398.77.

Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. Cambridge, MA: MIT Press.

Martens, J., & Sutskever, I. (2011). Learning recurrent neural networks with Hessian-free optimization. In L. Getoor & T. Scheffer (Eds.) *International Conference on Machine Learning* (pp. 1033–1040). Bellevue, WA:ACM.

McClelland, J. L. (2013). Integrating probabilistic models of perception and interactive neural networks: A historical and tutorial review. *Frontiers in Psychology*, *4*(August), 503. doi: 10.3389/fpsyg.2013.00503.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: Connectionist and dynamical systems approaches to cognition. *Trends in Cognitive Sciences*, *14*(8), 348–356. doi: 10.1016/j.tics.2010.06.002.

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

McClelland, J. L., Mirman, D., Bolger, D. J., & Khaitan, P. (2014). Interactive activation and mutual constraint satisfaction in perception and cognition. *Cognitive Science*, *38*, 1139–1189. doi: 10.1111/cogs.12146.

McClelland, J. L., & Plaut, D. C. (1999). Does generalization in infant learning implicate abstract algebra-like rules? *Trends in Cognitive Sciences*, *3*(5), 166–168.

McMullen, E., & Saffran, J. R. (2004). Music and language: A developmental comparison. *Music Perception: An Interdisciplinary Journal*, *21*(3), 289–311.

Mnih, A., & Hinton, G. E. (2007). Three new graphical models for statistical language modelling. In Z. Ghahramani (Ed.) *International Conference on Machine Learning* (pp. 641–648). New York: ACM Press. doi: 10.1145/1273496.1273577

Neal, R. M. (1995). *Bayesian learning for neural networks*. Ph.D. Thesis, Department of Computer Science, University of Toronto.

Nerbonne, J., & Heeringa, W. (2009). Measuring dialect differences. In E. J. Schmidt & P. Auer (Eds.), *Language and space: Theories and methods*. Berlin: Mouton De Gruyter.

Nerbonne, J., & Stoianov, I. (2004). Learning phonotactics with simple processors. In D. Gilbers, M. Schreuder and N. Knevel (Eds.), *On the boundaries of phonology and phonetics* (pp. 89–121). Groningen: University of Groningen.

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in Cognitive Sciences*, *2*(11), 455–462. doi: 10.1016/S1364-6613(98)01241-8.

Pasa, L., & Sperduti, A. (2014). Pre-training of recurrent neural networks via linear autoencoders. *Advances in Neural Information Processing Systems*, *27*, 3572–3580.

Pasa, L., Testolin, A., & Sperduti, A. (2015). Neural networks for sequential data: A pre-training approach based on Hidden Markov Models. *Neurocomputing*. doi:10.1016/j.neucom.2014.11.081

Patel, A. D. (2003). Language, music, syntax and the brain. *Nature Neuroscience*, *6*(7), 674–681. doi: 10.1038/nn1082.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco: Morgan Kaufmann.

Peña, M., Bonatti, L. L., Nespor, M., & Mehler, J. (2002). Signal-driven computations in speech processing. *Science*, *298*(5593), 604–607. doi: 10.1126/science.1072901.

Perruchet, P., Tyler, M. D., Galland, N., & Peereman, R. (2004). Learning nonadjacent dependencies: No need for algebraic-like computations. *Journal of Experimental Psychology. General*, *133*(4), 573–583. doi: 10.1037/0096-3445.133.4.573.

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: The CDP+ model of reading aloud. *Psychological Review*, *114*(2), 273–315.

Peterson, C., & Anderson, J. R. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, *1*(5), 995–1019.

Plaut, D. C. (1999). A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science*, *23*(4), 543–568. doi: 10.1207/s15516709cog2304_7.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*(1), 56–115.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*(1), 77–105.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In L. Bottou & M. Littman (Eds.) *International Conference on Machine Learning* (pp. 873–880). New York: ACM. doi: 10.1145/1553374.1553486

Rao, R. P. N., Olshausen, B. A., & Lewicki, M. (Eds.). (2002). *Probabilistic models of the brain: Perception and neural function*. Cambridge, MA: MIT Press.

Rastle, K., Harrington, J., & Coltheart, M. (2002). 358,534 nonwords: The ARC nonword database. *The Quarterly Journal of Experimental Psychology. A, Human Experimental Psychology*, *55*(4), 1339–1362.

Romberg, A. R., & Saffran, J. R. (2010). Statistical learning and language acquisition. *Wiley Interdisciplinary Reviews. Cognitive Science*, *1*(6), 906–914. doi: 10.1002/wcs.78.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations*. (Vol. 1). Cambridge, MA: MIT Press.

Saffran, J. R. (2001). Words in a sea of sounds: The output of infant statistical learning. *Cognition*, *81*(2), 149–169.

Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, *274*(5294), 1926–1928.

Saffran, J. R., & Wilson, D. P. (2003). From syllables to syntax: Multilevel statistical learning by 12-month-old infants. *Infancy*, *4*(2), 273–284. doi: 10.1207/S15327078IN0402_07.

Salakhutdinov, R., Tenenbaum, J. B., & Torralba, A. (2011). Learning to learn with compound HD models. *Advances in Neural Information Processing Systems*, *23*, 2061–2069.

Sang, E. F. T. K., & Nerbonne, J. (1999). Learning simple phonotactics. In T. Dean (Ed.) *Proceedings of the workshop on neural, symbolic, and reinforcement methods for sequence processing, ML2 workshop at IJCAI* (Vol. 99, pp. 41–46). San Francisco, CA: Morgan-Kauffman.

Seidenberg, M. S., MacDonald, M. C., & Saffran, J. R. (2002). Does grammar start where statistics stop? *Science*, *298*(5593), 553–554. doi: 10.1126/science.1078094.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, *7*(2–3), 161–193. doi: 10.1007/BF00114843.

Share, D. L. (1995). Phonological recoding and self-teaching: Sine qua non of reading acquisition. *Cognition*, *55*(2), 151–218; discussion 219–26.

Sibley, D. E., Kello, C. T., Plaut, D. C., & Elman, J. L. (2008). Large-scale modeling of wordform learning and representation. *Cognitive Science*, *32*(4), 741–754. doi: 10.1080/03640210802066964.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart & J. L. McClelland (Eds.) *Parallel distributed processing: Explorations in the microstructure of cognition, volume 1: Foundations* (pp. 194–281). Cambridge, MA: MIT Press.

Smolensky, P. (2006). Harmony in linguistic cognition. *Cognitive Science*, *30*(5), 779–801. doi: 10.1207/s15516709cog0000_78.

Spiegler, S., Golenia, B., & Flach, P. (2010). Unsupervised word decomposition with the Promodes algorithm. *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, *1*, 625–632.

Stoianov, I. (1998). Tree-based analysis of simple recurrent network learning. In C. Boitet & P. Whitelock (Eds.) *Proceedings of the 36th annual meeting on Association for computational linguistics* (Vol. 2, pp. 1502–1504). Morristown, NJ: Association for Computational Linguistics. doi: 10.3115/980691.980820

Stoianov, I. (1999). Recurrent autoassociative networks. In L. R. Medsker & L. C. Jain (Eds.), *Recurrent neural networks: Design and application*., (pp. 205–242). New York: CRC Press.

Stoianov, I. (2001). *Connectionist lexical processing*. Groningen Dissertation in Linguistic GRODIL 31.

Stoianov, I., & Zorzi, M. (2012). Emergence of a "visual number sense" in hierarchical generative models. *Nature Neuroscience*, *15*(2), 194–196. doi: 10.1038/nn.2996.

Sutskever, I. (2013). Training recurrent neural networks. Ph.D. Thesis, Department of Computer Science, University of Toronto.

Sutskever, I., Hinton, G. E., & Taylor, G. (2008). The recurrent temporal restricted Boltzmann machine. *Advances in Neural Information Processing Systems*, *20*, 1601–1608.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In L. Getoo & T. Scheffer (Eds.) *International Conference on Machine Learning* (pp. 1017–1024). Bellevue, WA: ACM.

Taylor, G., Hinton, G. E., & Roweis, S. (2011). Two distributed-state models for generating high-dimensional time series. *Journal of Machine Learning Research*, *12*, 1025–1068.

Testolin, A., Sperduti, A., Stoianov, I., & Zorzi, M. (2012). Assessment of sequential boltzmann machines on a lexical processing task. In M. Verleysen (Ed.) *European symposium on artificial neural networks, computational intelligence and machine learning* (pp. 275–280). Bruges: ESANN.

Testolin, A., Stoianov, I., De Filippo De Grazia, M., & Zorzi, M. (2013). Deep unsupervised learning on a desktop PC? A primer for cognitive scientists. *Frontiers in Psychology*, *4*, 251.

Tieleman, T. (2010). Gnumpy: An easy way to use GPU boards in Python. Technical Report UTML TR 2010-002, University of Toronto.

Toro, J. M., & Trobalón, J. B. (2005). Statistical computations over a speech stream in a rodent. *Perception & Psychophysics*, *67*(5), 867–875.

Ziegler, J. C., Perry, C., & Zorzi, M. (2014). Modelling reading development through phonological decoding and self-teaching: Implications for dyslexia. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *369*.

Zorzi, M., Testolin, A., & Stoianov, I. (2013). Modeling language and cognition with deep unsupervised learning: A tutorial overview. *Frontiers in Psychology*, *4*(August), 515. doi: 10.3389/fpsyg.2013.00515.