Assessment of Sequential Boltzmann Machines on a Lexical Processing Task

Alberto Testolin[1,2], Alessandro Sperduti[1], Ivilin Stoianov[2] and Marco Zorzi[2] *

1- Department of Pure and Applied Mathematics
2- Department of General Psychology and Center for Cognitive Sciences
University of Padova - Italy

Abstract. Recently, a promising probabilistic model based on Boltzmann Machines, i.e. the Recurrent Temporal RBM, has been proposed. It is able to learn physical dynamics (e.g. videos of bouncing balls), however up to now it was not clear whether this ability could apply to symbolic tasks. Here we assess its capabilities on learning graphotactic rules from a set of English words. It emerged that the model is able to extract local transition rules between items of a sequence, but it does not seem to be suited to encode a whole word.

1 Introduction

Several methods for dealing with temporal data have been proposed by the machine learning community [1]. In this work we will focus on connectionist models, whose application in this scenario was already discussed by J. Elman in his landmark paper on simple recurrent neural networks (SRN) [2]. Since then, many extensions and refinements on connectionist models have been developed, in order to deal with even more complex domains, where data can be highly structured [3].

The aim of this paper is to assess the capabilities of a recently introduced probabilistic graphical model based on Boltzmann Machines [4], which is able of manipulating sequential data through recurrent connections and it is therefore called *Recurrent Temporal Restricted Boltzmann Machine* (RTRBM, from now) [5]. It has some peculiar characteristics that make it interesting, not only from an engineering point of view but also for applications in computational cognitive modelling. First, the learning process is completely unsupervised because the network only learns to reproduce the training data as accurately as possible. We can therefore use it as a *generative model*, in order to produce new sequences that have a similar structure of those seen before. Moreover, learning exploits only local information and it is therefore more biologically plausible than classical backpropagation methods. Boltzmann Machines are experiencing a renaissance during last years, thanks to improved learning algorithms that allow an efficient training of large networks and that have been the basis for the development of promising paradigms like deep learning.

Thus far, RTRBM has been tested on motion capture, demonstrating that the network is capable to successfully extract the physical dynamics of such phenomena. Although such visual sequences are high-dimensional and present

*This study was supported by the European Research Council (grant no. 210922 to M.Z.)

high-level dependencies, their dynamic is generally smooth. Here we study the performance of the model on a symbolic task. The network was trained on a set of English words, presenting one letter at each time, thus assessing if the model is able to extract the *graphotactic rules* of that language, that are the compositional rules that describe how letters should be combined together in order to form plausible words. We compared the RTRBM ability of predicting the next letter of a word with other baseline learning algorithms in computational linguistics: n-gram models and Hidden Markov Models (HMMs). We also analysed the internal representations of the model (i.e. hidden units activations) in order to verify if the network was able to produce static, holistic representations of whole sequences. It emerged that the model principally extracts local transition rules instead of memorizing the entire sequence. Therefore, although the network obtains good performance on predicting the next element of a sequence, our study also points out some of its limitations.

2 The Recurrent Temporal Restricted Boltzmann Machine

An RTRBM is a partially directed graphical model with recurrent connections, defined in such a way that at each timestep hidden units activations depend both on the observed visible units (v) and on the previous hidden units (h) activations. A graphical representation of such a model is given in Fig. 1, where the network is unrolled over time in order to highlight sequential relations. RTRBMs are an extension of the well-known Restricted Boltzmann Machines, which define probability distribution over pairs of vectors exploiting a constrained graph structure that allows to factorize conditional distributions over variables.

The joint distribution induced by an RTRBM is defined as:

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

where the factor $P_0(v_1)P_0(h_1|v_1)$ corresponds to the probabilities associated with the first element of the sequence, when no previous context is available and therefore we use an initial bias b_{init}. If we know the current visible values v_t and the previous hidden values h_{t-1}, the new hidden activations are computed as:

$$P(H_t|v_t, h_{t-1}) = \sigma(VH^\top v_t + HHh_{t-1} + b_H) \quad (1)$$

where σ is the sigmoid function, VH is the matrix of visible-to-hidden weights, HH is the matrix of hidden-to-hidden weights and b_H is the vector of hidden units biases. Eq. 1 represents a mean field approximation, in which we consider the average of the neural activations instead of their stochastic correlations. Since we can compute the hidden units activations using this deterministic process, it turns out that inference in RTRBMs is very efficient, given the values of visible units, because we only have to sequentially compute hidden activations using Eq. 1. If we know the current hidden units activations h_t, the conditional

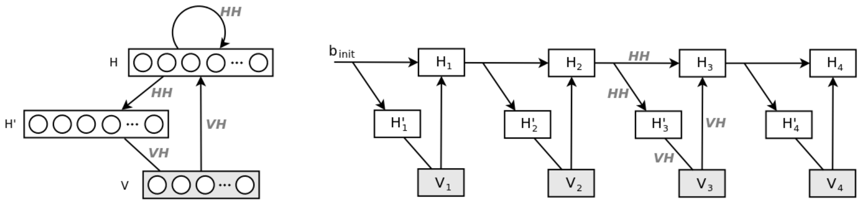

Fig. 1: Two schemes illustrating a Recurrent Temporal RBM.

distribution of the binary hidden units and the visible units at the following timestep is defined as:

$$P(V_{t+1}, H'_{t+1}|h_t) = \frac{\exp\left(v_{t+1}^{\top} VH h'_{t+1} + v_{t+1}^{\top} b_V + h'^{\top}_{t+1}(b_H + HH h_t)\right)}{Z(h_t)} \qquad (2)$$

where the factor $(b_H + HH h_t)$ represents the new biases for the binary hidden units of the RBM at time $t+1$, and it is computed taking into account the hidden unit bias b_H and the dynamic bias $HH h_t$ generated from the hidden units activations at the current timestep. Z is the so-called *partition function* and it is used to normalize values into legal probabilities.

According to Eq. 1 and Eq. 2, we can define a generative process that allows to get samples from the model distribution:

$$for\ 1 \leq t \leq T : \{sample\ v_t \sim P(V_t|h_{t-1});\ set\ h_t \leftarrow P(H_t|v_t|h_{t-1})\}$$

where the symbol $\sim$ indicates the sampling operation performed with block Gibbs sampling, while the symbol $\leftarrow$ stands for the deterministic assignment obtained using the mean field approximation. When generating the values of visible units, we thus need to use an MCMC algorithm, while once we have the visible units activations and the previous hidden units activations we can compute the new hidden units activations in just one step.

3 Lexical Processing Task

The focus of our work was on the lexical level of written language, hence one sequence corresponded to an English word. Previous research on phonotactic learning exploited simple recurrent networks as neural models [6] and demonstrated the effective capability of these systems to extract phonotactic rules from a given set of data. Here we aimed at exploring the potential of the RTRBM on the similar task of graphotactic learning, thus demonstrating that such a model is capable of extracting these rules from experience, without needing an explicit encoding of them or any prior knowledge about the task. Another desirable feature that a sequences neural processor should exhibit is the capability of developing rich holistic representations that correspond to whole sequences of elements. When manipulating temporal information, the network should gradually create an internal description that will eventually represent the information as a whole. In other words, the model should be able to encode dynamic information in a proper way such that we can perform further manipulations on it directly over the internal (possibly static and distributed) representations, instead of having to analyse the initial, external form of the data.

3.1 Method

The dataset used contained a large set of English monosyllables, thus almost exhaustively describing their graphotactic rules. Each letter was codified as a fixed-length binary vector using an orthogonal representation, hence the visible layer consisted of 27 units (one for each letter plus one for a termination symbol).

Weights were randomly initialized to small values and the learning rate was set to 0.025 and gradually decreased as the learning proceeded. The number of steps performed by the Contrastive Divergence procedure was scheduled to be small during the first phase of the training and successively increased (from 5, to 10, 25 and finally 40). Training stopped after 1200000 weight updates. We first trained an RTRBM with 110 hidden units over a small subset of 300 words (with lengths between 3 and 5) and then tested the scaling capabilities of the model by training another network with 200 hidden units over the complete dataset (5300 words for training and 1700 for testing, with lengths between 3 and 7).

In order to reduce the computational time required by learning and generative processes, we exploited NVIDIA graphic cards using the Gnumpy library [7] and adopting a mini-batch learning strategy, obtaining a speed-up of about 25 times.

We first evaluated the performance of the network on making predictions about the $(t+1)$-th element of a sequence, given the previous t elements. In other words, the model estimated the conditional probability of generating each letter, given the evidence represented by the current context. These probabilities represent the *successor distribution* associated with a certain context and they should be as close as possible to the empirical successor distribution computed on the training data [6]. We measured the prediction error by averaging the Euclidean distances between the vectors of model expectations and empirical distributions calculated for every possible prefix in the dataset. We then compared performance of RTRBM with other two families of statistical models: n-gram models, implemented as simple look-up tables where each row contains

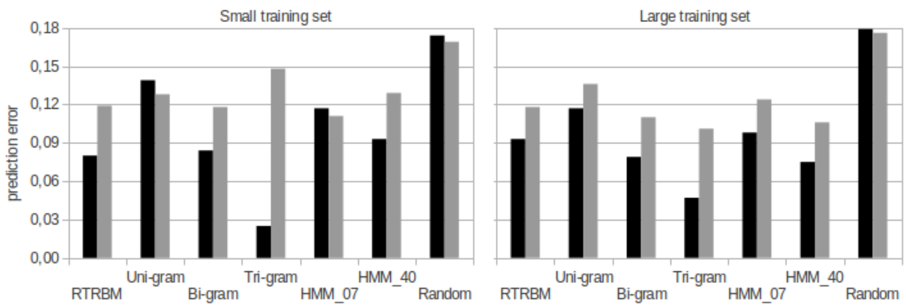

Fig. 2: Prediction errors on the training set (black) and on the test set (grey).

the successor distribution extracted from training data for each possible context (i.e. the last n letters analysed, with n varying between 1 and 3) and HMMs, trained according to a previous work on phonotactic learning [8] using 7 and 40 hidden states. The second metric adopted to evaluate the model consisted in testing its generative performance. We therefore collected a fixed number of samples ($s = 100000$) and calculated the *accuracy* (i.e. the ratio between generated sequences that were present in the training set and s) and the *completeness* (i.e. the ratio between generated sequences that were present in the training set and the total size of the training set) of the generation.

4 Results

Fig. 2 reports prediction errors for each model analysed. RTRBM obtained good performance over the small dataset (comparable to the one obtained by the bi-gram model), while its generalization ability over the large dataset did not improve as it happened for the other models. Fig. 3 shows that both the sampling indicators improve as the training proceeds. Note that, since the number of samples was kept equal for both the small and the large dataset, in the latter case the completeness value is lower. Nevertheless, sampling a greater number of sequences (e.g. one billion) resulted in a completeness of 80%.

Analysis of the internal (i.e., hidden layer) representations, generated after the production of the last letter of a word, revealed that the similarity between them (calculated as Euclidean distances) is correlated with the similarity between the corresponding sequences (measured with the Levenshtein distance), with a correlation coefficient r of 0.39 (see Fig. 4). It is important to note that the generative process can produce multiple instances of the same sequence; therefore, one crucial question is whether these different instances are associated to the same internal representations , or at least by highly similar representations. In particular, the similarity between internal representations for all instances of a certain sequence should be smaller than the similarity between internal representations across different sequences. As shown in Fig. 4, the internal representations corresponding to different words ("between class similarity") are

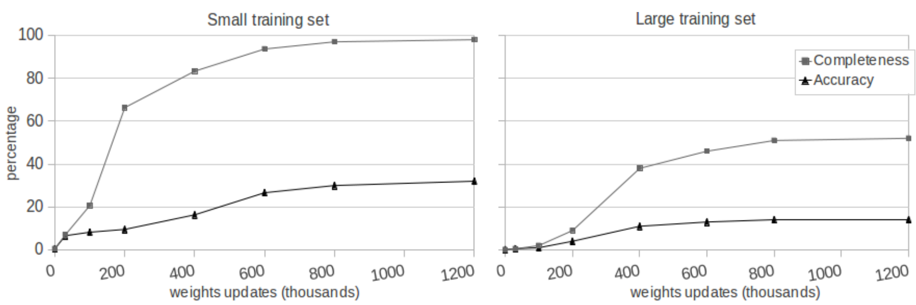

Fig. 3: Sampling completeness and accuracy collected during training.

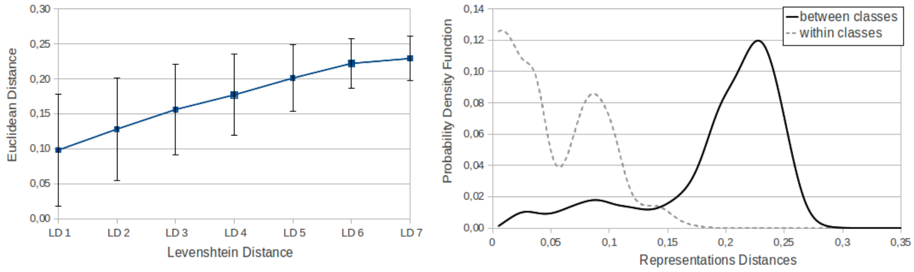

Fig. 4: Correlation between internal representations similarity and Levenshtein distances of corresponding words (left). Probability density functions of Euclidean distances between internal representations (right).

much more distant than the representations corresponding to different instances of the same word ("within class similarity"). Though the two distributions appear separated, the overlap between the respective tails suggests that perfect discrimination is not possible.

5 Conclusions and Future Directions

In this paper, we evaluated the performance of the Recurrent Temporal RBM model on learning sequences of letters corresponding to English words. Our results demonstrate that the network is able to learn local transition probabilities between sequence elements, that is graphotactic rules of the language, although its prediction ability does not fully match the performance of other state-of-the-art algorithms. Our study also points to a potential limitation of the model, because its internal representations do not seem to encode the entire sequence in a way that allows perfect discriminability between different sequences.

References

[1] T. Dietterich. Machine learning for sequential data: a review. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 227–246, 2009.

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

[3] A. Micheli, D. Sona, and A. Sperduti. Contextual processing of structured data by recursive cascade correlation. *Neural Networks, IEEE Transaction on*, 15(6):1396–1410, 2004.

[4] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[5] I. Sutskever, G.E. Hinton, and G. Taylor. The recurrent temporal restricted boltzmann machine. *Advances in Neural Information Processing Systems*, 21, 2009.

[6] J. Nerbonne and I. Stoianov. Learning phonotactics with simple processors. *On the Boundaries of Phonology and Phonetics*, pages 89–121, 2004.

[7] T. Tieleman. Gnumpy: an easy way to use gpu boards in python. *Department of Computer Science, University of Toronto*, 2010.

[8] E.F.T.K. Sang and J. Nerbonne. Learning simple phonotactics. In *Proceedings of the Workshop on Neural, Symbolic, and Reinforcement Methods for Sequence Processing', ML2 workshop at IJCAI*, volume 99, pages 41–46. Citeseer, 1999.