NETWORK MECHANISM SUPPORTING LONG-DISTANCE-DEPENDENCIES

Alexis Dubreuil
Institut de la Vision
Sorbonne Universités, INSERM, CNRS
F-75012 Paris, France
alexis.dubreuil@inserm.fr

Arthur Leblois
Institut des Maladies Neurodégénératives
Université de Bordeaux, CNRS, UMR 5293
33076 Bordeaux, France
arthur.leblois@u-bordeaux.fr

***Abstract*—Sequential behaviors such as language or bird songs are structured in time. This structure relies on the notion of long-distance-dependencies: transitions between words depend on the identity of words produced in the past. Here we propose a network mechanism supporting such dependencies. To do so we trained artificial neural networks to produce a minimal set of sequences exhibiting long-distance-dependencies. By reverse-engineering the trained networks we found this to rely on two superposing neural sequences, one responsible for the production of the motor sequence and another one encoding a contextual memory. We show how these two sequences are supported by neural activity and network connectivity and how they interact with each other to decide on transitions between words. We discuss similarities between the neural activity of our artificial neural networks and neural correlates of long-distance-dependencies that have recently been exposed in songbirds.**

I. Introduction

Long-distance-dependencies are a ubiquitous feature of temporally structured behavior. For instance in language the sentence "The cat near the trees catches the mouse" exhibits a long-distance-dependency where the number of the subject (singular) conditions the number of the verb that appears latter in the sentence [1], [2]. They are also a characteristic of animal behavior, in particular songbirds [3], where their neural correlates have begun to be exposed [4].

Here we propose to take advantage of the accessibility of artificial neural networks, that are known to be able to extract long-distance-dependencies from data [5], and reverse-engineer them to expose a neural network mechanism that supports such long-distance-dependencies. To do so we trained recurrent-neural-networks (RNN) to produce a minimal set of sequences (see Fig. 1a). It consists of two sequences of words where the first word (A or A') sets a context that later decides on the last word of the sentence (E or E'), with a common non-ambiguous sub-sequence in the middle (B-C-D). Such a behavior relies on three cognitive components: the production of a motor sequence, the memorization of the context throughout the sequence, and the biasing of the motor sequence by the memory. By reverse-engineering the trained RNN, we show how network connectivity supports the neural activity underlying each of these three cognitive components. Throughout this reverse-engineering process, we expose multiple shared commonalities with neural recordings in a premotor brain nucleus of songbirds (HVC, used as a proper name) involved in controlling song syntax. In some instances we ascribe a functional role to peculiar neural activity features through causal circuit manipulations. Finally we discuss how our findings relate to previous results obtained in other reverse-engineering studies that dealt with sequential behavior.

II. Methods

A. Task and network architecture

Here we consider recurrent neural networks (RNN) that receive inputs from two neurons and can produce outputs through seven readout neurons (Fig. 1b). At the beginning of each task trial, the network is cued by one of the two input neurons (square pulse of amplitude 1 of duration 100ms, Fig. 1c). As a response to such a cue, the network's output should consist of one of two sequences (sequence 1 and 2) of five consecutive readout activations, each with a square profile of amplitude 1 and a duration of 1000ms (Fig. 1d). The two readout sequences are chosen as a minimal set of sequences involving long-distance-dependencies: the first activation is sequence dependent (activation of the A or A' readout neuron), the following third activations are common to both sequences and the last activation is sequence dependent (activation of the E or E' readout neuron, Fig. 1a).

The recurrent architecture consists of $N = 256$ neurons whose activation functions are rectified-linear-units ($\Phi(.) = \text{ReLU}(.)$). The dynamics of the input currents $\vec{x} \in \mathbb{R}^N$ to neurons obeys

$$\tau \frac{d\vec{x}}{dt} = -\vec{x}(t) + W^{rec}\Phi(\vec{x}(t)) + W^{in}\vec{u}(t) + \vec{\eta}(t) \quad (1)$$

$\vec{u} \in \mathbb{R}^2_+$ models the activity of input neurons, $W^{in} \in \mathbb{R}^{N\times 2}$ contains the two connectivity vectors connecting the input neurons to the recurrent neurons. $W^{rec} \in \mathbb{R}^{N\times N}$ connects recurrent neurons that are assigned a time constant $\tau = 100$ms. Each entry of $\vec{\eta}(t) \in \mathbb{R}^N$ is an independent white noise whose variance is specified below. Recurrent activity is read out by seven readout neurons with activity $\vec{z} \in \mathbb{R}^7_+$:

$$\vec{z} = W^{out}\Phi(\vec{x}) \quad (2)$$

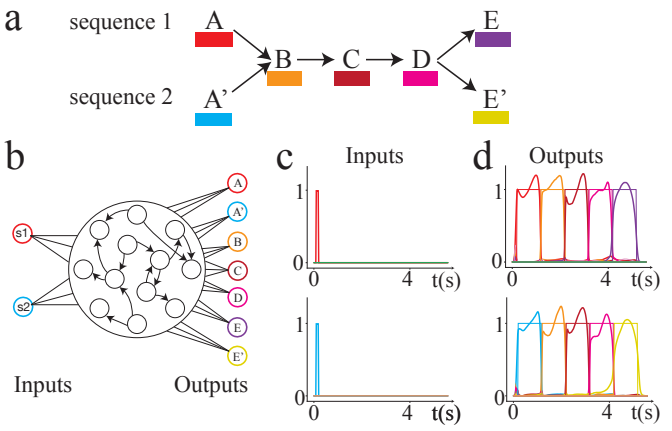

Fig. 1. a. Two sequences with long-distance-dependencies. For clarity, in subsequent figures each word is associated with a color, as in [4]. b. Cartoon of the trained neural architecture. c. One of two input neurons is transiently activated at the beginning of each sequence. d. Each of the output neuron is responsible for producing one of the 7 possible words of the sequence. Each colored line represents the activation of one of the readout neurons (thin lines represent the target activations, thick lines represent activations from trained networks).

where $W^{out} \in \mathbb{R}^{7\times N}$ connects recurrent neurons to the readout neurons. W^{out} is chosen to connect each readout neuron to only a subset of 30 recurrent neurons (with a weight of 1), with non-overlapping subsets across readout neurons, such that a total of 210 neurons are connected to a readout neuron, and 46 are not connected to any readout neuron. This fixed readout architecture is chosen to model the descending pathway from HVC to the Robust nucleus of the arcopallium (RA) in songbirds, that we here consider as non-plastic.

B. Network training

For network training, the continuous dynamics is discretized to

$$\vec{x}^{t+1} = (1-\alpha)\vec{x}^t + \alpha\left(W^{rec}\Phi(\vec{x}^t) + W^{in}\vec{u}^t\right) + \sqrt{2\alpha\sigma^2}\vec{\mathcal{N}}(0,1) \tag{3}$$

with $\alpha = \frac{\Delta t}{\tau} = 0.2$ and $\sigma = 0.01$ to parametrize the amplitude of the individual neuron noise term. Training is performed with custom codes, which include the non-standard leak term $(1-\alpha)\vec{x}^t$, written in pytorch using the ADAM optimizer with standard parameters (learning rate is 0.001, decay rates for the first and second moments are 0.9 and 0.999 respectively) [7]. This algorithm minimizes a quadratic loss which measures the distance between the activity of the readout neurons at time t in trial k, $\vec{z}^{k,t}$ and the target activations $\vec{\hat{z}}^t$, and that includes a L1-norm regularization term:

$$\mathcal{L} = \frac{1}{K\times T\times 7}\sum_{k=1}^{K}\sum_{t=1}^{T}\left(|\vec{z}^{k,t} - \vec{\hat{z}}^t|^2 + \frac{\lambda}{N}\sum_{i=1}^{N}|\Phi(x_i^{k,t})|\right) \tag{4}$$

$T = \frac{1000\text{ms}}{\Delta t}$ denotes the number of time steps that constitute a sequence. We trained networks for various values of λ and analyzed networks successfully trained with the highest value of the regularization parameter, $\lambda_{max} \in [0.02, 0.05]$. Minimization is performed on the connectivity matrices W^{rec} and W^{in} with individual matrix entries initialized as $W^{rec}_{ij} = \frac{w}{\sqrt{N}}$ and $W^{in}_{ij} = w$ where w stands for the realization of a standard Gaussian random variable. Training is performed on batches of $K = 60$ trials that mix the sequences 1 and 2. Each trial is initialized with $\vec{x}^{t=0} = \vec{0}$ and a newly drawn independent neural noise. Below we present the reverse-engineering of one of the trained networks, we have checked on two other networks that using the same learning protocol leads to networks relying on the same mechanism.

III. Results

Here we expose a network mechanism for the implementation of long-distance-dependencies that we uncovered by reverse-engineering an artificial neural network trained to produce the two sequences of Fig. 1a. In the first subsection we show how sequential neural activity and network connectivity support the production of the non-ambiguous part of the motor sequence, B-C-D. In the second subsection we expose how the memory of the first word A or A' is maintained in network's activity throughout the sequence A/A'-B-C-D. In the third section, we show how this memory switches the direction of the motor sequence either towards the word E or the word E'.

A. Neural implementation of sequential motor activity

Here we focus on characterizing the neural activity supporting the production of the subsequence B-C-D. The structure of the population activity can be mapped onto the produced motor sequence, with groups of neurons connected to a readout neuron being active during the production of the corresponding word (Fig. 2a). Most of the neurons firing rates are maintained active throughout an entire word production, with a temporal profile that typically ramps up (Fig. 2b) or down (Fig. 2c) from the word onset, as corroborated by Fig. 2e which shows the distribution of the time of peak firing rates. In our trained RNN, we thus observe ramping activity locked to word boundaries, a tendency also observed in the HVC of bengalese finch producing courtship songs [10]. Another subset of neurons show a much more temporally focused neural activity (Fig. 2c), being active for 100ms precisely at word boundaries. Such a neuron active at the boundary between words X and Y is typically connected to neither readout X or Y and thus do not participate in the activation of relevant readout neurons. Rather, inactivation of these neurons show that their functional role is to control the timing at which neural activity transitions from the production of one word to another. This is illustrated in Fig. 2g, where we show the effect of inactivating these neurons that are active at the transition between words B and C. It does not influence the sharpness with which the transition occurs but lengthens the duration of word B (from 1000ms without inactivation to 1600ms with inactivation) and increases the variability of the duration of word B (standard deviation from 22ms to 131ms, this variability is due to the neural noise injected in each neuron, see Methods) as shown in Fig. 2h.

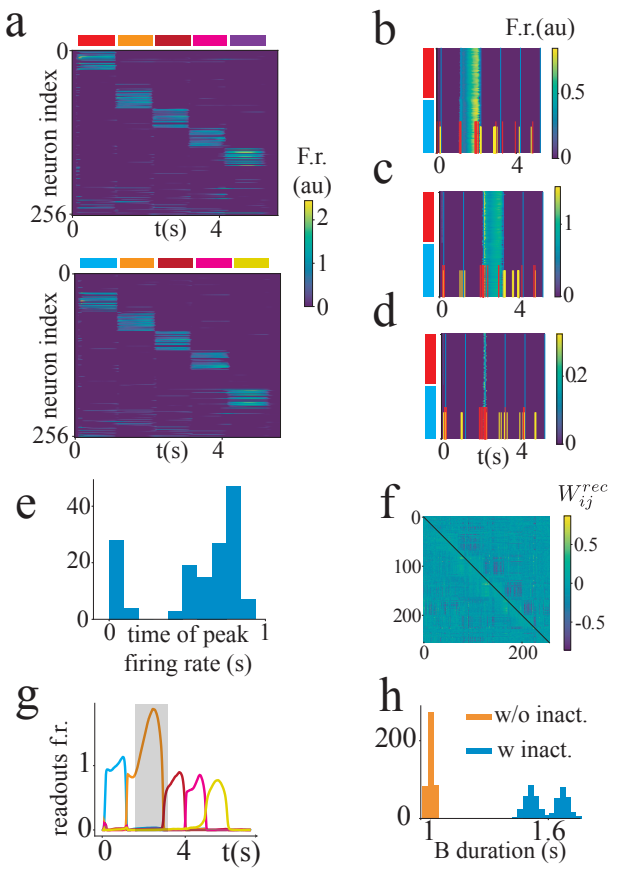

Fig. 2. a. Neural activity of the entire population during the production of sequence 1 (top) and sequence 2 (bottom). Neurons are ordered according to how they are connected to the seven readouts (see Methods), with 30 neurons associated to each readout (ordered as A,A',B,C,D,E,E') and the 46 neurons not connected to any readout at the bottom. b,c,d. Firing rate of three example neurons over multiple renditions of the sequence, the first 50 lines correspond to sequences 1 starting by the red-labeled word A, the next 50 lines correspond to sequences 2 that start by the blue-labeled word A'. Red lines (respectively yellow lines) show the time of peak-firing rate of the 20 most excitatory (respectively inhibitory) pre-synaptic neurons. e. Distribution of peak firing rates for neurons that connect to a readout and that have their peak firing rate within the timing boundaries of their readout (154 neurons out of 210), 0 and 1s stand for the beginning and end of words. f. Connectivity matrix of the network, here neurons have been ordered according to their time of peak firing rate in sequence 1. g. Example of activations of the readout neurons while burst neurons usually active between the orange and burgundy words are inactivated. h. Distribution for the duration of the orange word when burst neurons are inactivated (orange) or not (blue).

This sequential activity is supported by the network connectivity shown in Fig. 2f. To plot this matrix, neurons have been reordered according to their time of peak firing rate in sequence 1. It shows a form of block structure, with neurons active in a word inhibiting neurons that where active in the previous word. Excitation seems to be mostly provided by the early firing neurons of the group. This feature if further illustrated by the red (respectively yellow) lines in Fig. 2b,c,d, which show for the three example neurons the time of peak firing rate of their 20 most excitatory (respectively inhibitory) pre-synaptic neurons.

B. Neural implementation of the contextual memory

For the single neuron examples from Fig. 2b,c,d activity is almost not modulated by the context set by the first word A or A'. While this is the case for most neurons, we also observed neural activities that are modulated by the context, as for the neuron shown in Fig. 3a. To get a sense of how the memory of the context is encoded in the activity of the whole network, we trained a decoder to predict the context A or A' and extracted the neural features on which the prediction relies. More precisely, for each time step, a perceptron is trained to classify patterns of $N = 256$ firing rates, with one class corresponding to firing rate vectors obtained when the first word is A (the output of the decoder is trained to be $+1$), and A' for the other class (the output of the decoder is trained to be -1). This type of simple decoder successfully recovered the context (Fig. 3b) and an analysis of the trained weight-vectors showed that the memory of context is supported by different neurons throughout the motor sequence (Fig. 3c). For each time step, we extracted the five neurons that contribute the most to a class A prediction (the five most positive entries of the weight-vector) and the five neurons that contribute the most to a class A' prediction (the five most negative entries of the weight-vector). In Fig. 3d we show how the neural activity of these class A or class A' neurons is modulated by context. It thus confirms the intuition obtained from the single neuron example Fig. 3a, that the memory of context is represented as a sequence of modulations of the firing rate of neurons that are also involved in producing the motor sequence. Indeed, most of the class A and A' neurons are connected to one of the readout neuron (74%). This sequence of firing rate modulation appears to be supported by network's connectivity. To show this, in Fig. 3e we plotted the connectivity matrices between subsets of class A (top) and class A' (bottom) neurons, with neurons ordered by increasing timing of maximal contextual modulation, such that an entry above the diagonal represents the connection W_{ij} from a neuron j whose activity is maximally modulated at a time t_j to a neuron i whose activity is maximally modulated at a time $t_i > t_j$. Similarly to the motor sequence studied in the previous section, we found that for both connectivity matrices, the averaged connectivity from later to earlier modulated neurons is inhibitory (averaged connectivity of -0.05 and -0.08 above the diagonal), while from earlier to later modulated neurons it is excitatory (averaged connectivity of $+0.03$ and $+0.01$ below the diagonal).

C. Contextual memory biases motor sequence

These sequences of firing rate modulations, that we have shown are encoding the memory of context, are causally involved in biasing the motor sequence towards E or E'. To show this we inactivated the subset of five class A or A' neurons that are most modulated at a specific time point. We performed such inactivations at six time points equally spaced throughout the sequence B-C-D. It systematically lead to a wrong choice for the last word as illustrated in Fig. 4a for a specific inactivation. To understand how the memory

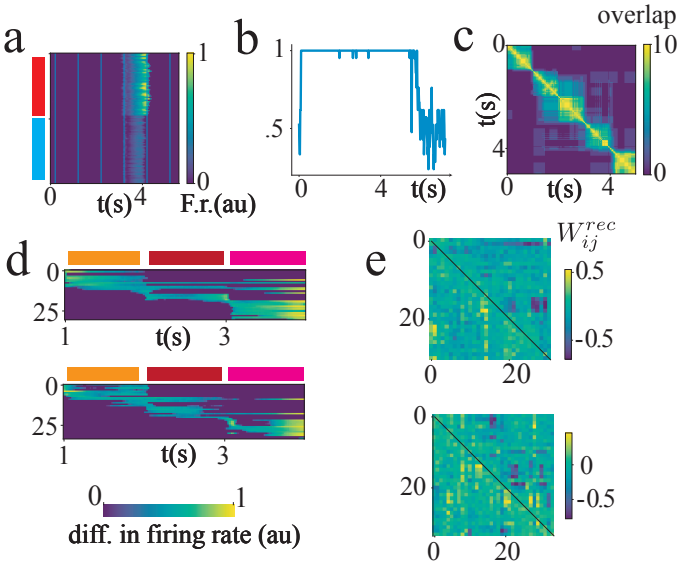

Fig. 3. a. Example of a neuron, connected to the readout D whose firing rate is modulated by the identity of the first red or blue word. b. Decoding performance of perceptrons trained to decode the first word from the firing rates of the 256 neurons at all times in the sequence. c. Cross-correlogram of decoding vectors: for each time point the perceptron is parametrized by a weight-vector, for each of these weight-vectors the 10 entries corresponding to the 10 neurons that participated the most in the first word prediction (highest absolute value) are set to 1, other entries to 0, to obtain the decoding vector. Color bar shows the scalar product between two decoding vectors obtained at different decoding time points. d. Rescaled differences in firing rates for the class A (top) and class A' (bottom) neurons. The color amplitude represents $log(1+100*[fr1-fr2]_+)$ with fr1 and fr2 the firing rates produced during sequence 1 and 2 and $[]_+$ the sign rectification function. e. Connectivity matrix of the class A (top) and A' (bottom) neurons. Neurons are ordered according to their time of maximal firing rate modulation.

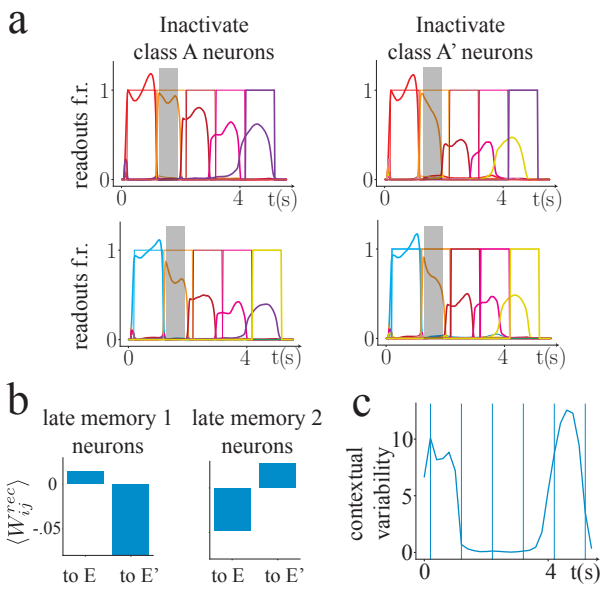

Fig. 4. a. Example of the firing rates of readout neurons, while inactivating (for the period materialized by the shaded area) the five neurons that encode the most memory A (left) or A' (right) at 1.3s. b. Averaged outgoing connectivity for the five neurons that most encode memory A 100ms before the transition from D to E or E'. Only connectivity towards neurons connected to the E or E' readouts are shown. c. Context-to-context variability for the whole population activity, obtained by squaring, for each neuron, the difference between the firing rates in the two sequences, and averaging over neurons.

actually biases the motor sequence, we focused on the neurons that most encode the memory of context A just before the ambiguous transition D-E or D-E'. It turns out that these neurons, who exhibit an increased firing rate in context A, tend to excite neurons that are connected to the readout E and to inhibit neurons that are connected to the readout E' (Fig. 4b). The converse is true for neurons that most encode the memory of context A' just before the transition. It is also interesting to note, that for the memory to bias the motor sequence, the modulations of firing rates with context are largely increased right before the ambiguous transition as compared to non-ambiguous transitions (Fig. 4c), as observed in recordings from the HVC nucleus of songbirds [4].

IV. Discussion

We trained a RNN to solve a minimal task exhibiting long-distance dependencies. The RNN produces sequences of words. The first word sets a context which is memorized throughout the sequence. Later in the sequence this memory is taken into account to decide the last word produced (Fig. 1A). To access a corresponding network mechanism, we reverse-engineered the trained RNN, i.e. we characterized how neural activity relates to the produced sequence of words and how network connectivity relates to this neural activity.

We described this network mechanism relying on three components. The first component supports the production of a sub-sequence of words with unambiguous transitions between them. The corresponding neural activity is sequential and relies on asymmetric excitations and inhibitions where a neuron active at time t in the sequence tends to excite a neuron active at a later time $t + \Delta T$ while it is inhibited by this neuron, as observed in other RNN trained to produce sequential outputs [12]–[14]. A particularity of our sequences is the rather long duration of words (1s for a 100ms neural time constant), as a consequence, excitation and inhibition tend to be organized in blocks and neural activity shows peculiar selectivity profiles as discussed in more details below. The second component of the network mechanism is the encoding of the contextual memory. In our setting, this memory takes the form of a sequence of firing rate modulations that superimposes onto the neural sequence producing the network's output. A similar type of neural sequences superposition has also been described in RNN for the concurrent representation of temporal and non-temporal information [13]. The third component of the network mechanism supports the ability of the contextual memory to switch the motor sequence towards a particular word. Neurons that encode a particular contextual memory right before an ambiguous word transition tend to excite neurons responsible for the production of the next word associated with this context and to inhibit neurons that produce the other possible word.

Another reverse-engineering study has focused on a task involving long-distance-dependencies, a linguistic number agreement task where words are presented to a RNN and it is examined how the RNN outputs produce the next word of the sentence with a correct number agreement [2]. They reported similarities with the mechanism exhibited here, in particular regarding the third component of the mechanism, where contextual memory neurons excite (inhibit) the neurons coding for the correct (incorrect) next word. The second component of the network mechanism is rather different with two highly specialized units encoding the equivalent of the contextual memories A and A' and showing persistent activity throughout the sequence. It would be interesting to see whether such persistent activity could emerge, and under what conditions, in a vanilla RNN as used here [14] or whether this relies on the gates available in the more elaborated LSTM architecture used in [2].

Similarly to other reverse-engineering studies (see e.g. [15]–[18]), the neural activity of our RNN shares common properties with the neural activity observed in brains of behaving animals. By training to produce sequences of rather long words, we found neurons with ramping activity profiles, a feature similar to recordings in songbirds showing such ramping activity locked onto syllable boundaries [10]. Another type of selectivity that we observed in the RNN are boundary neurons that are active over a very short time right at a boundary between words. We showed that these neurons control the timing of the transitions between words. These neurons share similarities with boundary neurons that have been observed in mammals [9] and in songbirds [11] (but see [19], [20] for discussions regarding the pro-eminence of such selectivity profiles). As for the neural basis of the contextual memory maintenance, recent recordings in the HVC of canaris have shown that modulation of firing rate by context are circumscribed around a few time points, with no reporting of individual neural activity persisting throughout long-periods of time. The extent to which motor related neural sequences and context memory sequences overlap in the HVC nucleus is not known, but our results suggest that those two can in principle be implemented by the same neural network, without resorting to several brain areas. Finally, has also observed in the HVC of canaris, the context-to-context variability of firing rates is very low at time points where transitions between words are unambiguous and increases before ambiguous transitions (Fig. 4c, [4]). We have thus spotted multiple commonalities between the neural activity in our RNN and in the HVC of songbirds, it would be interesting to enrich trained sequences with more realistic features and see whether we can obtain a more exhaustive and quantitative picture of HVC's physiology. Long-distance-dependencies are at the core of all structured sequential behaviors and it would be interesting to see whether our neural mechanism could be at play in more complex tasks such as language processing. The possibility to perform high-density electrocorticography recordings in humans performing language tasks [21] can in principle allow to test whether long-distance-dependencies are supported by a similar mechanism in human and bird brains.

To conclude, reverse-engineering RNN trained on a minimal task allowed us to access a network mechanism for the implementation of long-distance-dependencies. It would be interesting to train networks on sequences with more elaborated structures. It would, first, allow to access mechanisms for richer syntactic phenomenon and second, to better model the rich phenomenology exposed by neural recordings from animals performing structured sequential behaviors such as songbirds.

REFERENCES

[1] Linzen, T., Dupoux, E., and Goldberg, Y. "Assessing the ability of LSTMs to learn syntax-sensitive dependencies". Transactions of the Association for Computational Linguistics, 4:521– 535, 2016

[2] Lakretz, Y. et al. "The emergence of number and syntax units in LSTM language models". Proceedings of NAACL, Minneapolis, MN, 2019.

[3] Markowitz, J. E., Ivie, E., Kligler, L. and Gardner, T. J. "Long-range Order in Canary Song". PLoS Computational Biology 9, e1003052 (2013).

[4] Cohen, Y. et al. "Hidden neural states underlie canary song syntax". Nature 582, 539–544, 2020.

[5] Hochreiter, S. and Schmidhuber, J. "Long short-term memory". Neural Computation, 9(8):1735-1780, 1997.

[6] Song, H. F., Yang, G. R. and Wang, X.-J. "Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible Framework." PLOS Computational Biology 12, e1004792, 2016.

[7] Kingma, D.P. and Ba, J. "Adam: A method for stochastic optimization". arXiv preprint arXiv:1412.6980, 2014.

[8] Hahnloser, R. H. R., Kozhevnikov, A. A. and Fee, M. S. "An ultra-sparse code underliesthe generation of neural sequences in a songbird". Nature 419, 65–70, 2002.

[9] Barnes, T. D. et al. "Advance cueing produces enhanced action-boundary patterns of spike activity in the sensorimotor striatum". Journal of Neurophysiology 105, 1861–1878, 2011.

[10] Fujimoto, H., Hasegawa, T. and Watanabe, D. "Neural Coding of Syntactic Structure in Learned Vocalizations in the Songbird". Journal of Neuroscience 31, 10023–10033, 2011.

[11] Amador, A., Perl, Y. S., Mindlin, G. B. and Margoliash, D. "Elemental gesture dynamics are encoded by song premotor cortical neurons". Nature 495, 59–64, 2013.

[12] Rajan, K., Harvey, C. D. and Tank, D. W. "Recurrent Network Models of Sequence Generation and Memory". Neuron 90, 128–142, 2016.

[13] Bi, Z. and Zhou, C. Understanding the computation of time using neural network models. Proceedings of the National Academy of Sciences 117, 10530–10540, 2020.

[14] Orhan, A. E., and Ma, W. J. "A diverse range of factors affect the nature of neural representations underlying short-term memory". Nature neuroscience, 22(2), 275-283, 2019.

[15] Mante, V., Sussillo, D., Shenoy, K. V. and Newsome, W. T. "Context-dependent computation by recurrent dynamics in prefrontal cortex". Nature 503, 78–84, 2013.

[16] Sussillo, D., Churchland, M. M., Kaufman, M. T. and Shenoy, K. V. "A neural network that finds a naturalistic solution for the production of muscle activity". Nature Neuroscience 18, 1025–1033, 2015.

[17] Chaisangmongkon, W., Swaminathan, S. K., Freedman, D. J. and Wang, X.-J. "Computing by Robust Transience: How the Fronto-Parietal Network Performs Sequential, Category-Based Decisions". Neuron 93, 1504-1517.e4, 2017.

[18] Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F. and Ostojic, S. "Complementary roles of dimensionality and population structure in neural computations". bioRxiv, 2020.

[19] Lynch, G. F., Okubo, T. S., Hanuschkin, A., Hahnloser, R. H. R. and Fee, M. S. "Rhythmic Continuous-Time Coding in the Songbird Analog of Vocal Motor Cortex". Neuron 90, 877–892, 2016.

[20] Picardo, M. A. et al. ”Population-Level Representation of a Temporal Sequence Underlying Song Production in the Zebra Finch”. Neuron 90, 866–876, 2016.
[21] Moses, D. A., Leonard, M. K., Makin, J. G. and Chang, E. F. ”Real-time decoding of question-and-answer speech dialogue using human cortical activity”. Nature Communications 10, 3096, 2019.