ROBUSTNESS OF TWO-LAYERED FEEDFORWARD NEURAL NETWORK

Maja Sarevska

European University -
Republic of Macedonia (EURM)

Abstract:

In this paper a two-layer feedforward network is studied, a network that stores an association between two sequences in the two layers. Our work shows that neuronal sequences in one area can robustly trigger sequences in the second area if the association between the sequences is stored in the network. A more detailed incorporation into the biological aspects of neural network in the network dynamics may help to improve neutral networks in engineering applications.

Key words:

Robustness, Two-Layered Neural Network, Sequence Association.

1. INTRODUCTION

Replay of sequential activity patterns in the hippocampus has been proposed as a mechanism for the consolidation of episodic memories [1]. It is thought that replay sequences originate in one area and trigger neuronal sequences downstream, e.g., in other area, like neocortex [2]. However, under physiological conditions internal noise or external interference are likely to corrupt the precise sequential ordering of neuronal sequences. It remains an open question how robustly the activation of a corrupted sequence in one brain area can induce the associated sequence in the second area. Here we study this question in a two-layer feedforward network that stores the association between two sequences in the two layers. While keeping the connection weights fixed, we degrade the input sequence incrementally and observe the sequence induced in the output layer. We measure the similarity of sequences with the Spearman rank-order correlation. Surprisingly, we find that even when the input sequence is highly corrupted, the retrieved output sequence is similar to the associated sequence. This result is specific to the stored association and not found for random sequences. Our work shows that neuronal sequences in one area can robustly trigger sequences in a second area if the association between the sequences is stored in the network. Incorporating biological details of neural network improves robustness of association of neural network [3].

Correspondence:
Maja Sarevska

e-mail:
maja.sarevska@eurm.edu.mk

DOI: 10.15308/Sinteza-2017-50-54

2. MODEL

Network Dynamics

We use neural network model with two layers consisting of excitatory and inhibitory units [4]. These units represent populations of neurons and have heterogeneous parameters, unless otherwise stated in the Results. Excitatory units in the first layer send projections to excitatory (w_{E1E1}) and inhibitory (w_{E1I1}) units in the first layer, as well as to excitatory (w_{E1E2}) and inhibitory (w_{E1I2}) units in the second layer (Fig. 1). Within the second layer, there are recurrent excitatory connections (w_{E2E2}) and excitatory to inhibitory connections (w_{E2I2}). Here we only briefly sketch the model; the full description can be found in [4]. The dynamics of the activity xj of excitatory unit j in the first layer is given by

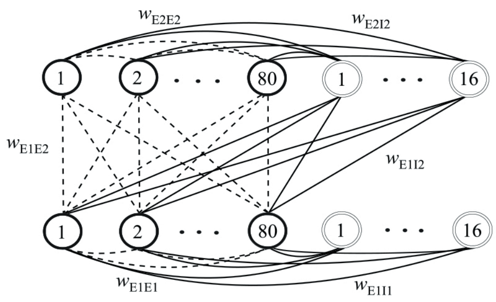

Fig. 1. Two-layered feedforward neural network

$$\begin{aligned}\tau_e\dot{x}_j = -x_j(t) + f\Bigg(& a_{ee}x_j(t) \\ &+ C\sum_{k=1}^{N} c_{jk}x_k(t-\Delta_{jk}) \\ &- a_{ei}\sum_{k=1}^{M} v_{jk}y_k - dz_j + e\eta_j - \theta_e + I_j\Bigg) \\ = -x_j(t) + v_j(t)&\end{aligned} \tag{1}$$

where τ_e is time constant of the excitatory unit, the dot represent the derivative with respect to time, *f* is sigmoid function, and *vj(t)* is the firing rate of population j. The seven terms in the function *f* represent different inputs to the unit. These are, firstthe self excitation of population *j* with its connectivity a_{ee}. Second, the inputs from other excitatory units with a delay of Δ_{jk} weighted by connection weights c_{jk} and a general excitability *C*. Third, inputs from inhibitory units with activity ya_k weighted by random static connection strength v_{jk}. and a general excitability from inhibitory units a_{ei}. Fourth, an adaptation current z_j weighted by a constant *d* with dynamics

$$\dot{z}_j = \frac{-z_j}{\tau_z} + b(1-z_j)v_j \tag{2}$$

where τ_z is the decay rate and *b* its rate of growth. The fifth term is noise defined by an Ornstein-Ulenbeck process

$$\dot{\eta}_j = \frac{-\eta_j}{\tau_\eta} + \xi_j \tag{3}$$

where ξ_j is white noise process with zero mean and unit standard deviation (std). The parameter τ_η allows us to adjust the noise level. The sixth term I_j represents external inputs. Finally, θ_e is the excitatory threshold.

Recurrent connections undergo spike-timing dependent plasticity (STDP)

$$\tau_c\dot{c}_{jk} = P_{STDP} - D_{STDP} - \psi \tag{4}$$

where τ_c is a time constant, P_{STDP} represents potentiation of the connection and D_{STDP}depression. The term ψ is a reducing factor, which provides stability to the dynamics. More details can be found in [4].

The dynamics of inhibitory units are described by

$$\dot{y}_j = \frac{-y_j}{\tau_i} + a_{ie}\left[\sum_{k=1}^{N} Q_{jk}x_k - \theta_i\right]_+ + iI_j \tag{5}$$

where a_{ie} represents the strength of the excitatory to inhibitory connections. The notation [x]+ is equivalent tomax(*x*,*0*). Q_{jk} is the static strength of individual excitatory to inhibitory connections, and θ_i is the threshold to activate inhibitory units.

The dynamics of excitatory units in the second layer are similar to the first layer with the addition of feedforward inputs

$$\begin{aligned}\tau_e\dot{x}_j = -x_j(t) + f\Bigg(& a_{ee}x_j(t) \\ &+ C\sum_{k=1}^{N} c_{jk}x_k(t-\Delta_{jk}) + g\sum_{k=1}^{N} K_{jk}\,\chi_k(t) \\ &- a_{ei}\sum_{k=1}^{M} v_{jk}y_k - dz_j + e\eta_j - \theta_e + I_j\Bigg) \\ = -x_j(t) + v_j(t)&\end{aligned} \tag{6}$$

where parameter *g* is a gain term, and K_{jk} is the strength of the connection from excitatory unit *k* in the first layer to excitatory unit *j* in the second layer. Plasticity between the layers is given by

$$\tau_K\dot{K}_{jk} = (\alpha - c_{jk})(\xi - c_{jk})\left[\Xi - \sum_{r=1}^{N} K_{jr}\chi_r(t)\right]_+ H_jH_k - \psi \tag{7}$$

where τ_k, α, ξ and Ξ are constants, and $H(x)$ is a sharp sigmoid functions. Detail on the reducing factor ψ can be found in [4].

Differential equations were solved using the Euler method with an integration time-step of 0.5 ms.The model parameters are defined in Table 1.

3. QUANTIFYING THE ROBUSTNESS OF SEQUENCE ASSOCIATION

In most our simulations, we stored associations between a sequence in the first layer, say units 1 to 80, and a sequence in the second layer, say units 1 to 20, in the connection weights of the network as described in [4]. We then examined the robustness of sequence association by applying perturbed sequences in the first layer (input sequences) and observing the evoked sequences in the second layer (output sequences). To quantify the differences between two sequences, we calculated the Spearman rank order correlation ρ between the activation times of the units in the two sequences. Since the units' activities are continuous variables, we had to define when the units are considered to have become active. Here we used the time of the first local maximum in a unit's activity. The correlation takes values between -1 and 1. A correlation of 1 or -1, implies that the two sequences are identical or reversed, respectively. The closer the correlation is to zero, the more the sequences differ from each other.

Layer I		Layer II	
Parameter	Value	Parameter	Value
a_{ee}	8	a_{ee}	6
a_{ei}	15	a_{ei}	15
a_{ie}	15	a_{ie}	15
b	0.05	b	0.05
C	3	C	2
d	11	d	12
e	1	e	1
N	80	N	80
M	16	M	16
θ_e	3.8	θ_e	3.9
θ_i	0.2	θ_i	0.2
τ_e	10	θ_{inc}	0.3
τ_c	70	τ_e	10
τ_i	5	τ_c	70
τ_z	70	τ_i	5
τ_η	5	τ_z	70
		τ_η	5
		τ_k	70
		α	0.01
		ξ	1
		Ξ	5

Table 1. Mean parameter values used in our simulations. Heterogeneous values were drawn from a normal distribution with standard deviation of 4%. The units for all time constants are milliseconds.

To study the robustness systematically, we examined the relationship between the output correlation and the input correlation (Fig. 2). By input and output correlation we mean the correlation between the test sequences and their respective reference sequences. If for large changes of input correlation we obtain low changes in the output correlation then the network is robust (gray line), for the opposite case we get non-robust network (black line). We therefore generated 650 input sequences with input correlations roughly uniformly distributed in [-1,1], applied them on the first layer and recorded the invoked sequence in the second layer. Calculating the correlation for given sequences is straightforward, but generating input sequences with a desired correlation to the reference input sequence is nontrivial. Since random sequences have predominantly small correlations with any reference sequence, we implemented a systematic search algorithm. To obtain large positive correlations, we started with the reference sequence and successively increased the number of permutations until a sequence within the desired range of correlation was generated. We followed a similar procedure for large negative correlations, except that we started with the reversed reference sequence. For correlations close to zero, we selected random sequences. We then fit a sigmoid function to the relationship between output and input correlations

$$F1(x) = \frac{2}{1 + be^{-cx}} - 1 \tag{8}$$

and quantified the robustness by the parameter c. The closer the curve is to the maximal robustness, i.e., a sign function, the larger the robustness parameter *c*. Equation (8) is convenient for fitting the data for robust sequence association, but is not appropriate for non-robust relationships. In these cases, we use anotherfitting function instead

$$F2(x) = d\tan(ax) \tag{9}$$

We performed model selection based on the Akaike Information Criterion (AIC) to decide in each case whether (8), (9), or a combination of both

$$F(x) = F1(x) + F2(x) \tag{10}$$

best accounted for the input-output relationship.

To examine whether the robustness of sequence association is specific to the stored sequences and not found for random sequences, we estimated the robustness parameter c for random reference sequences. We generated reference input sequences that had certain correlations with the input sequence used during storage. The correlations were divided in 10 bins of width 0.2. For each bin, we generated 40 randomized input sequences, which served as the reference input sequences. For each randomized input sequence, we recorded the output sequence generated by the network in the second layer. We then performed the robustness analysis described above with this pair of input and output reference sequences.

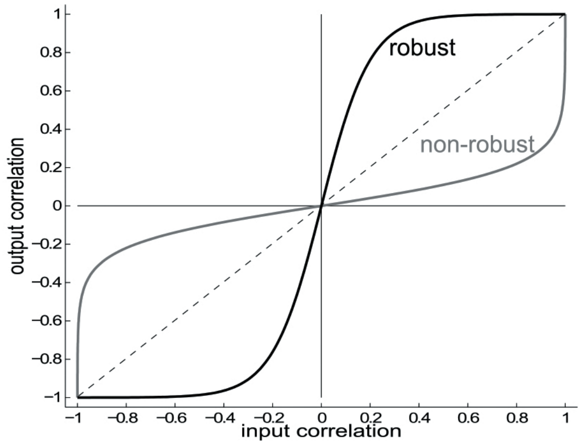

Fig. 2. Graphical presentation for network robustness

4. RESULTS

In the first analysis, we reduced the network dynamics to focus on the role of the feedforward connections between the two layers. We set the connection strengths wE2E2 to zero and switched off plasticity after storing the association between a sequence in the first layer and a sequence in the second layer. After training the feedforward connection strengths, using sequence 1÷80 and 1÷20 in the first and second layer respectively, we tested the sequence retrieval in the network with a corrupted input sequence. From the network, we retrieved an output sequence, which was very similar to the stored output sequence. So, a corrupted input sequence can retrieve the correct output sequence in this example. This is reminiscent of pattern completion, which is well-known in neural network models that store patterns of neural activity. The systematic analysis of the input-output function revealed that indeed the network robustly associated the input with the output sequence (Fig. 3a). This is the central result of this paper that we wish to study in more detail in the following.

To test whether the observed robustness is simply an artifact of our data analysis, we tested the robustness for deviant reference sequences. If robustness was a result of sequence association, robustness should be observed only for the sequences used during storage, and not for random ones. When we used a random reference sequence that bore no resemblance to the stored sequence, the network did not seem to “recognize“ the sequence and retrieved an output sequence that was quite different from the output sequence used during storage . Using these two reference sequences, the input-output relationship was not robust (Fig. 3b). When we examined the range of possible reference sequences, we found that robustness is highest for the stored sequence but drops steeply when the reference sequences deviate from the stored ones (Fig. 3c). The observed robustness is therefore a direct result of sequence association in the network connections.

5. CONCLUSIONS

In this paper the robustness of sequence association between two layers in a neural network is quantified by testing the input-output relationship of the network with distorted input sequences. We found that a purely feedforward network associates the input and output sequences robustly. There are, however, open questions and limitations to our study. For instance, when looking at all possible permutation of a reference sequence with many elements, the number of sequences with a correlation around zero is much larger than the number of correlations close to 1 or -1. By contrast, we took the view that we were probing the input-output relationship of the network and therefore used input sequences with roughly uniformly distributed input correlations. The difference in distribution might affect the parameter fits, and it is not immediately clear to us which distribution of input correlations should be preferred. Perhaps neither of the distributions discussed above should be used since they are both rather artificial scenarios. Perhaps, the best thing to do is to use the distribution of input sequences that is generated by the biological network. However, while some models have been proposed, none is widely accepted. We therefore opted to generate the input sequences according to the simple procedure used here to clearly isolate the effect of the feedforward connectivity on the robustness of sequence association. Future work will be needed to investigate the influence of the statistics of the input sequences on the robustness.

Another question is the potential influence of the procedure for assigning a unique time of activation to each unit's activity.

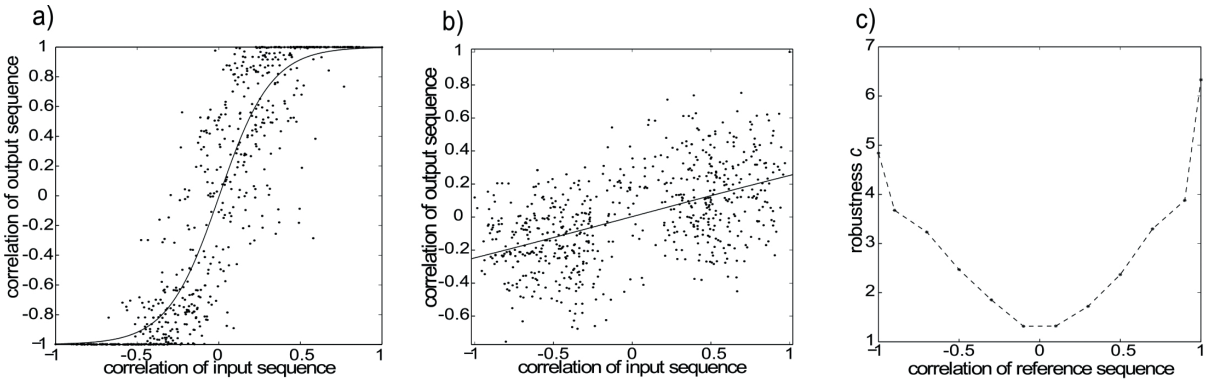

Fig. 3. Results: a) robust sequence association; b) robustness for random sequence; c) robustness for random sequences in range [-1.1]

Here we used the first local maximum of the activation. Other potential measures could have been used, such as the first time of crossing a certain threshold, the median time of the activity distribution, or the time of the absolute maximum. Since there is no universally agreed upon measure for rate-based units, it is speculative at this point. This ambiguity could be resolved by using spiking neuron models, where the sharp spikes allow a clear definition of times of activity.

We used only one fixed set of network parameters, including a relatively modest network size, since the network dynamics and analyses require extensive numerical calculations. Future work is needed to study robustness of sequence association for different network parameters, especially different network sizes and different level of inhibition.Additional analysis should be made for noise influence and influence of plasticity on final performances.

REFERENCES

[1] Scoville, WB Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, *20*, 11–21..

[2] Buonomano, D. V. (2003). Timing of neural responses in cortical organotypic slices. *Proceedings of the National Academy of Sciences of the United States of America, 100*(8), 4897–902. doi:10.1073/pnas.0736909100

[3] Lytton, William W., From Computer to Brain. The Foundations of Computational Neuroscience. Springer Science & Business Media. May, 2007.

[4] Verduzco-Flores, S. O., Bodner, M., & Ermentrout, B. (2012). A model for complex sequence learning and reproduction in neural populations. *Journal of computational neuroscience, 32*(3), 403–23. doi:10.1007/s10827-011-0360-x.