Learning Structure in Time with a Plastic Recurrent Neural Network

Fabian Schubert

April 16, 2018

1 Introduction

We implemented a neural network consisting of binary neurons, modeled in discrete time steps, which follows the ideas presented in [1]. The architecture of our network is depicted in Fig. 1 and shall be described in further detail.

The recurrent network consists of a population with $N_e = 300$ excitatory units (denoted as x_e) and a population with $N_i = 60$ (denoted as x_i) inhibitory units. Furthermore, a population of $N_{ext} = 9$ excitatory units (I_j) is interpreted as external input, where the input coming from each external unit is to be interpreted as encoding a particular feature of a sensory stream, e.g. the recognition of a particular letter or symbol.

2 Methods

2.1 Network Details

Synaptic connectivities - represented by arrows in the illustration - were initially generated from a uniform distribution and the following properties, listed in Table 1.

2.2 Neuron Model

The state of the neurons is updated in discrete time steps by the following equations:

Table 1: Network parameters

Connection Fraction W_{ee}	0.05
Connection Fraction W_{ei}	0.1
Connection Fraction W_{ie}	0.2
Connection Fraction W_{ii}	0.2
Connection Fraction $W_{e,ext}$	0.1
$\langle W_{e,ext} \rangle$	0.2
Total postsynaptic E→E input	1.0
Total postsynaptic I→E input	−1.0
Total postsynaptic E→I input	1.0
Total postsynaptic I→I input	−1.0

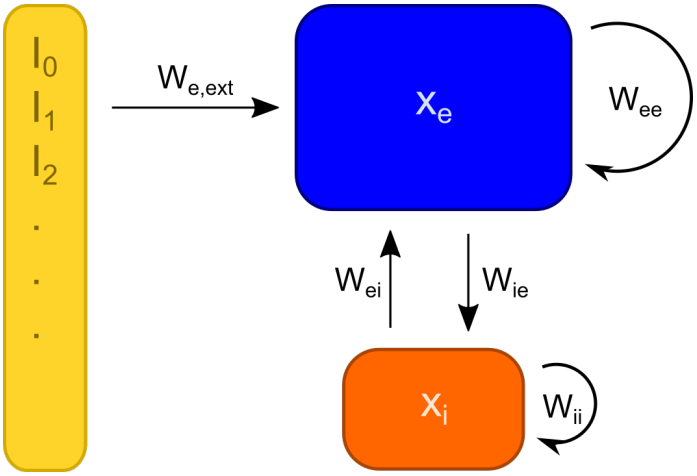

Figure 1: Architecture of the RNN

Table 2: Parameters of the neuron model

$\mu_{noise,e}$	0.0
$\mu_{noise,i}$	0.0
$\sigma_{noise,e}$	0.1
$\sigma_{noise,i}$	0.1
$\mu_{target,e}$	0.1
$\mu_{target,i}$	0.1
$\sigma_{target,e}$	0.0
$\sigma_{target,i}$	0.0
μ_{IP}	0.002

$$x_{e,n}(t+1) = \theta\left(\sum_{j=0}^{N_e-1} W_{ee,nj}x_{e,j}(t) + \sum_{k=0}^{N_i-1} W_{ei,nk}x_{i,k}(t) + \sum_{l=0}^{N_{ext}-1} W_{e,ext,nl}I_l(t) - T_{e,n}(t) + \xi_{e,n}(t)\right) \tag{1}$$

$$x_{i,n}(t+1) = \theta\left(\sum_{j=0}^{N_e-1} W_{ie,nj}x_{e,j}(t) + \sum_{k=0}^{N_i-1} W_{ii,nk}x_{i,k}(t) - T_{i,n}(t) + \xi_{i,n}(t)\right) \tag{2}$$

where $\theta(\cdot)$ is the theta function and T_e and T_i represent additional threshold values. $\xi_{e/i}$ are random noise terms sampled from a Gaussian distribution at each time step with parameters $\mu_{noise,e/i}$ and $\sigma_{noise,e/i}$.

To stabilize network activity, each neuron's threshold is updated each time step such that the neuron's average activity approach a given target value:

$$T_{e/i,n}(t+1) = T_{e/i,n}(t) + \mu_{IP}\left(x_{e/i,n}(t) - r_{target,e/i,n}\right) \tag{3}$$

where μ_{IP} is the learning rate of this "intrinsic plasticity". Target rates $r_{target,e/i}$ were drawn randomly from a Gaussian distribution with parameters $\mu_{target,e/i}$ and $\sigma_{target,e/i}$ for each neuron and kept fixed throughout the simulation.

Parameters of the dynamics described in this section are given in Table 2.

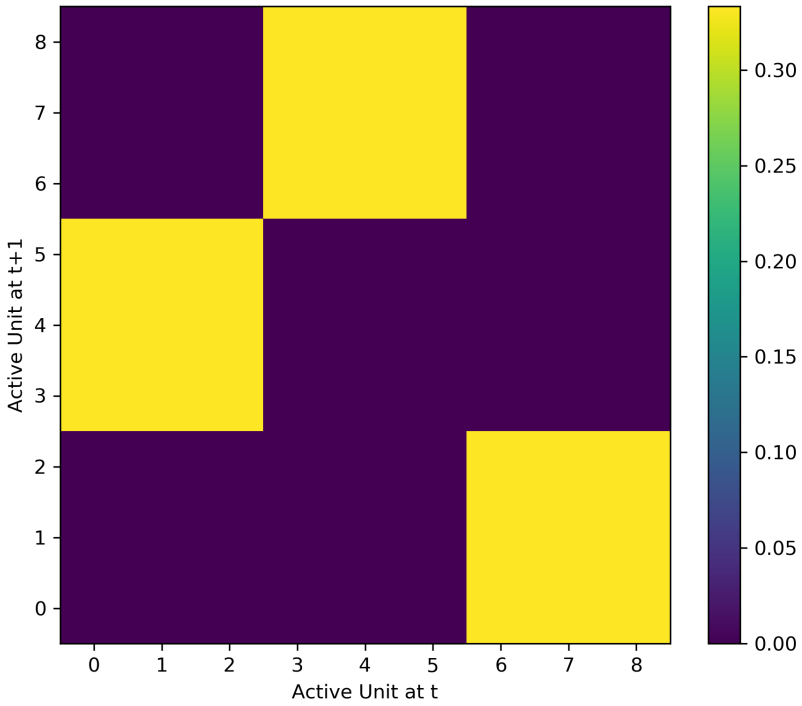

Figure 2: Transition Matrix between subsequently active input states

2.3 Rules of the Input Sequence

In each time step, only a single unit is in its active state $I_j(t) = 1$. The sequence of active input nodes was generated by a markov chain with transition probabilities shown in Fig. 2.

Due to the structure of the transition matrix, the sequence is partially predictable in the sense that an element of $\{0, 1, 2\}$ will always be followed by an element of $\{3, 4, 5\}$ etc.

2.4 Plasticity Rules

Recurrent excitatory connection were subject to two plasticity mechanisms: A simple pre-post Hebbian learning rule and a postsynaptic multiplicative normalization preventing connectivity runaway.

$$\Delta W_{ee,ij}(t) = \mu_{hebb} \left(x_{e,j}(t-1) x_{e,i}(t) - x_{e,i}(t-1) x_{e,j}(t) \right) \tag{4}$$

$$W_{ee,ij}(t) = w_{total,ee} \frac{W_{ee,ij}(t-1) + \Delta W_{ee,ij}(t)}{\sum_{k=0}^{N_e-1} W_{ee,ik}(t-1) + \Delta W_{ee,ik}(t)} \tag{5}$$

We did not include pruning or creation of synapses, but set a very small lower bound for existing excitatory connections.

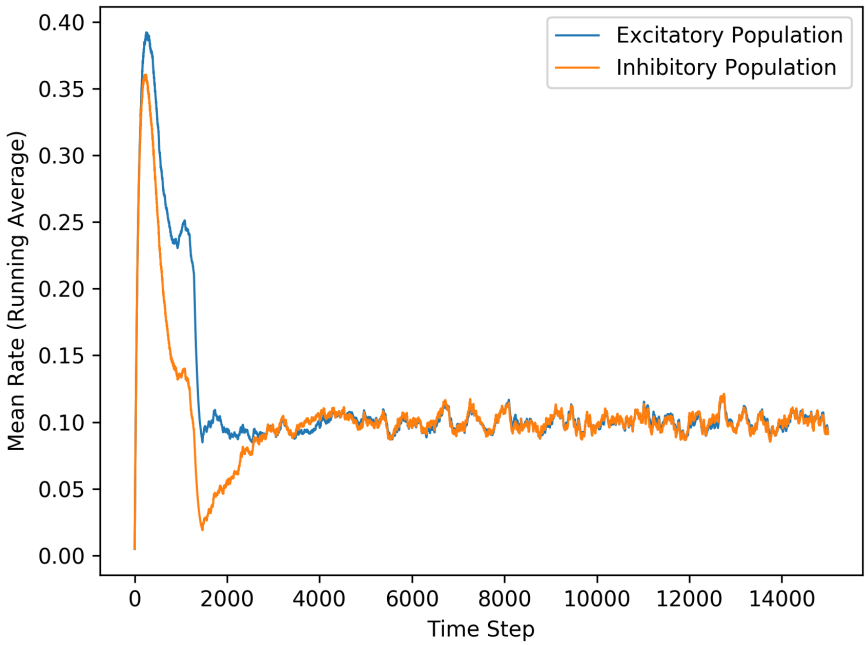

Figure 3: Running average of excitatory and inhibitory population activity.

3 Results

Network activity settled at a constant mean rate and a corresponding mean treshold, see Fig. 3 and Fig. 4.

Furthermore, Fig. 5 and Fig. 6 suggest that the appearance of active states follows poissonian statistics. However, the distribution of excitatory inter"spike"-intervals shows a clear preference for multiples of 3, which was not present in the absence of external input and is reflected in the 3-fold periodicity of the input sequence.

Generally speaking, the implemented plasticity rules often gave rise to time courses of synaptic weights similar to the one shown in Fig. 7: the emergence of one or more comparably strong weights alongside a majority of weak connections.

3.1 Cluster Analysis of Excitatory Activity

Following the conceptual idea presented by Elman [2], we performed a hierarchical cluster analysis of the binary activity vectors of the exitatory population. For this, we used the activity data of x_e from the last 3000 steps. We then performed a hierachical cluster analysis with Ward's method. The resulting dendrogram is depicted in Fig. 8. A 3-fold structure is visible in the uppermost branching layer.

4 More Complex Input Sequences

In the input sequence used in the previous section each letter corresponds to a a particular state in a Markov model. Thus, the prediction of the next letter only

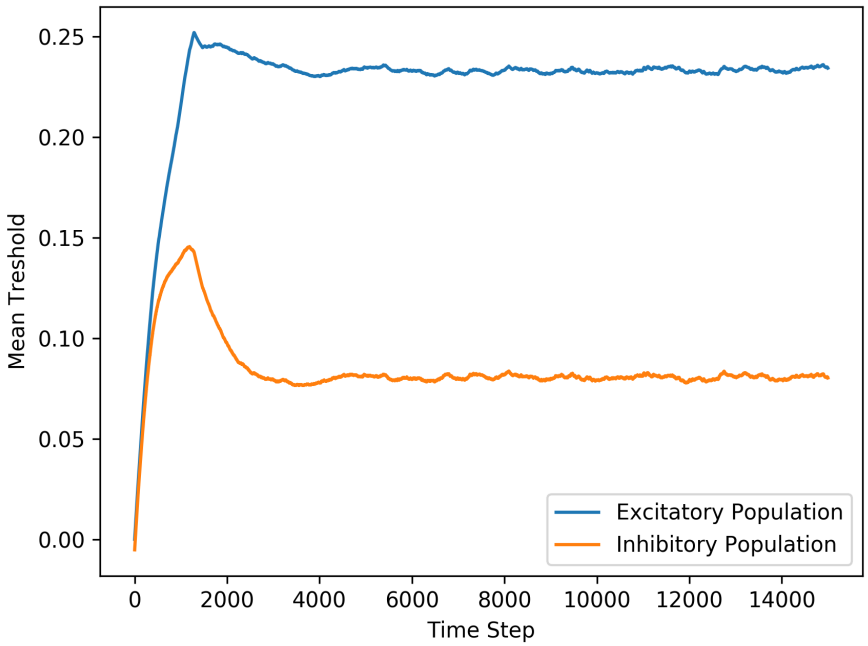

Figure 4: Population mean of excitatory and inhibitory thresholds.

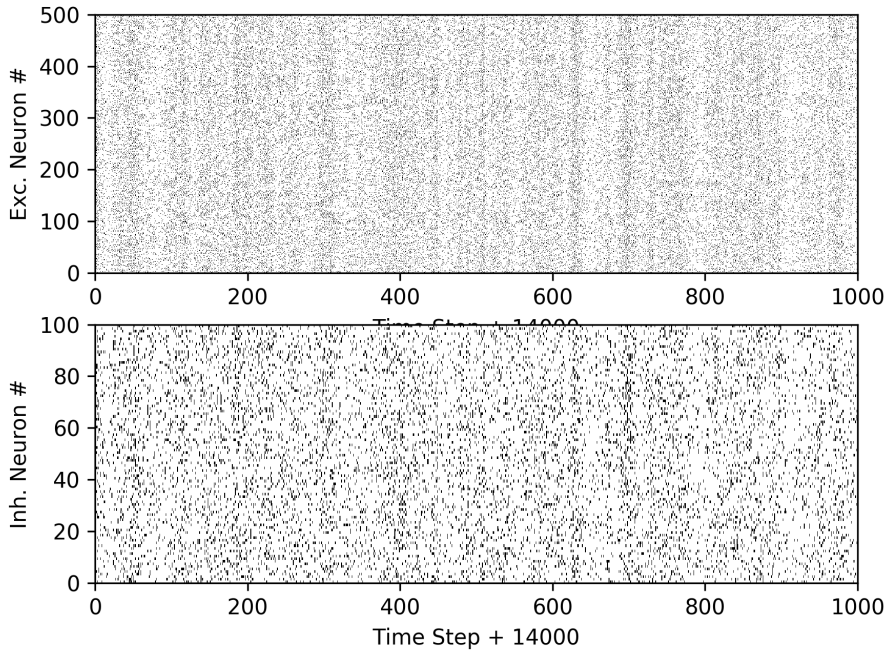

Figure 5: Raster plot of network activity.

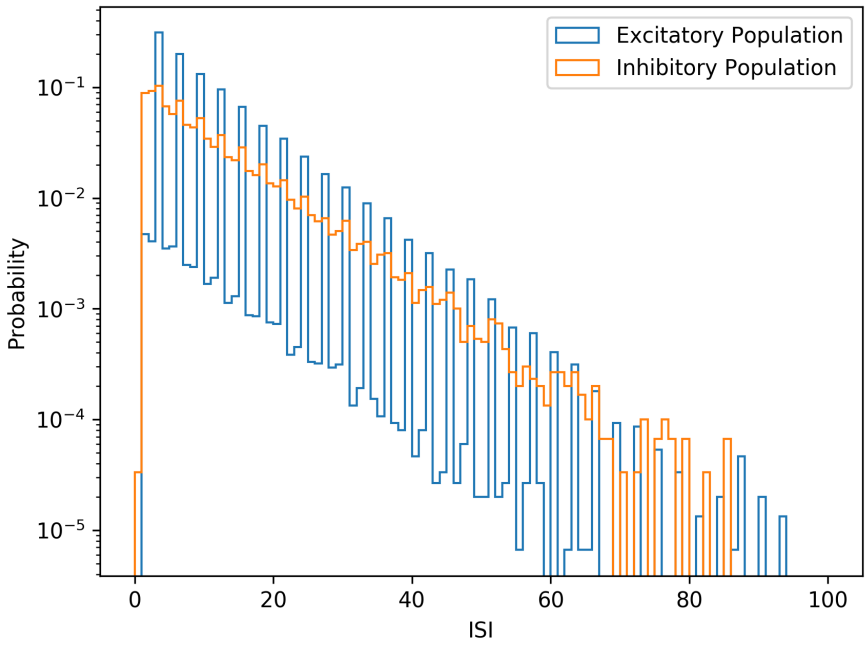

Figure 6: Distribution of interspike intervals.

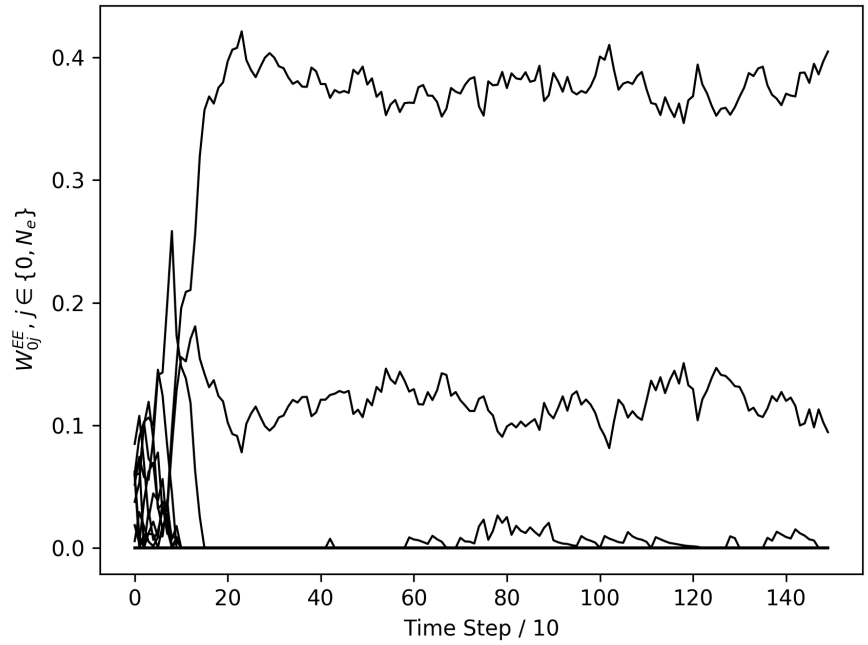

Figure 7: Sample time course of E-¿E weights.

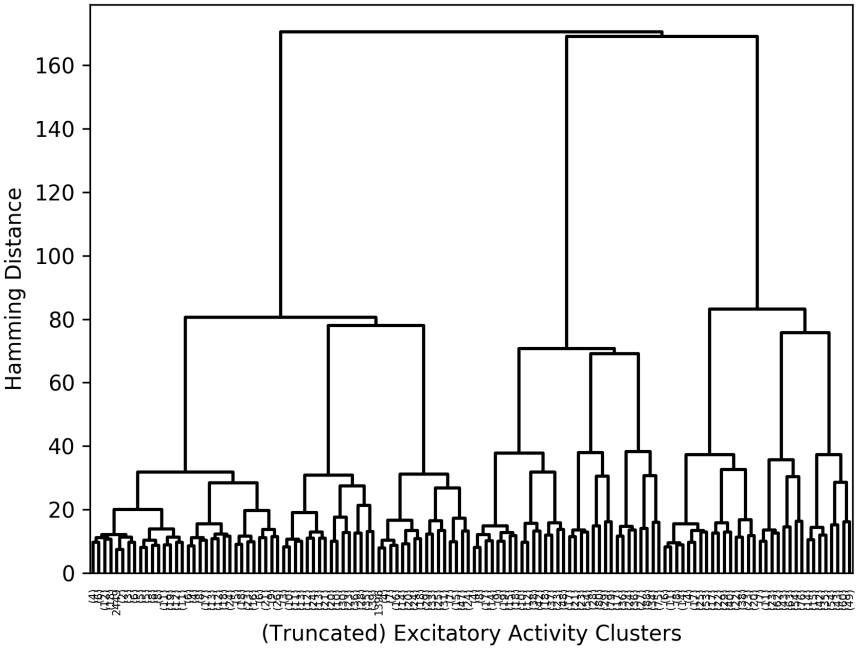

Figure 8: Dendrogram of a cluster analysis of excitatory activity patterns.

requires knowledge about the previous letter received. A more realistic kind of input stream would require the network to gather information about multiple time steps in order to optimally predict the next letter. Such a kind of artificial grammar was also used in [1] and initially introduced by Reber [3]. It is depicted in Fig. 9(top). Interestingly, the rules of this grammar is described as a finite-state machine where the *transitions* between states correspond to letters. Of course, this kind of representation of the dynamics also has a counterpart where letters are identified with states, see the bottom network in Fig. 9. Our previous analysis of network activity was based on the assumption that letters, or sets of letters were represented by states of the recurrent network and not in transitions between those. In consequence, a reflection of the Reber grammar within the network should resemble the mapping shown in the bottom network of Fig. 9.

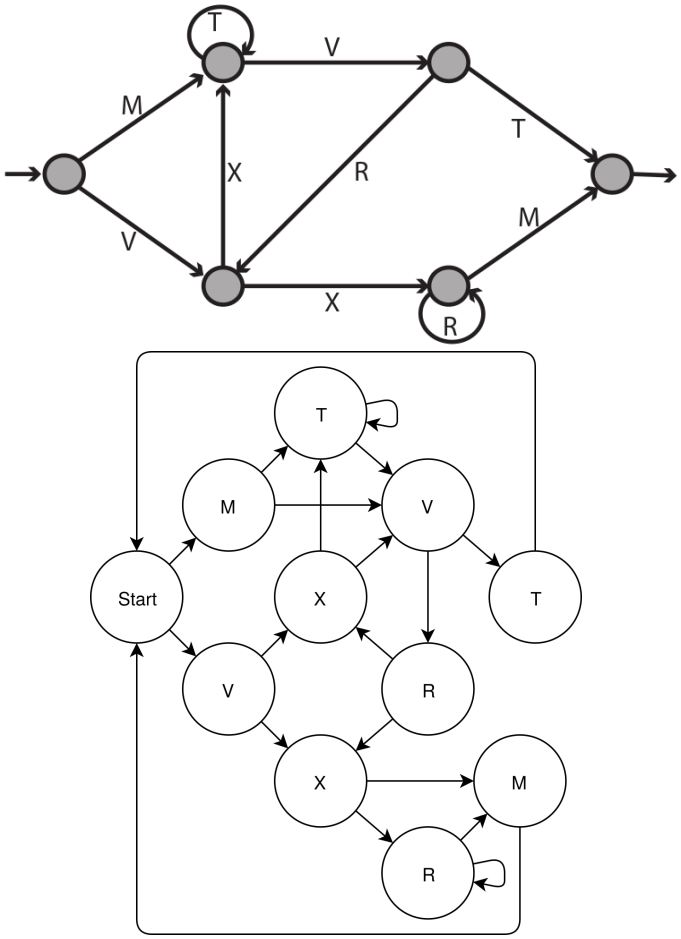

Figure 9: Artificial Grammar Rule proposed by Reber [3] (top) and the corresponding network where letters are represented by states (bottom).

References

[1] R. Duarte, P. Series, and A. Morrison. Self-Organized Artificial Grammar Learning in Spiking Neural Networks. In *36th Annual Conference of the Cognitive Science Society*, 07 2014.

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

[3] A. S. Reber. Implicit Learning of Artificial Grammars. *Journal of Verbal Learning and Verbal Behavoir*, 1967.