A Robust Model of Gated Working Memory

Anthony Strock[1,2,3], Xavier Hinaut[1,2,3,*], and Nicolas P. Rougier[1,2,3,*]

[1]INRIA Bordeaux Sud-Ouest, Bordeaux, France
[2]LaBRI, Université de Bordeaux, Institut Polytechnique de Bordeaux, Centre National de la Recherche Scientifique, UMR 5800, Talence, France
[3]Institut des Maladies Neurodégénératives, Université de Bordeaux, Centre National de la Recherche Scientifique, UMR 5293, Bordeaux, France
[*]Equal contribution

September 2, 2019

Abstract: Gated working memory is defined as the capacity of holding arbitrary information at any time in order to be used at a later time. Based on electrophysiological recordings, several computational models have tackled the problem using dedicated and explicit mechanisms. We propose instead to consider an implicit mechanism based on a random recurrent neural network. We introduce a robust yet simple reservoir model of gated working memory with instantaneous updates. The model is able to store an arbitrary real value at random time over an extended period of time. The dynamics of the model is a line attractor that learns to exploit reentry and a non-linearity during the training phase using only a few representative values. A deeper study of the model shows that there is actually a large range of hyper parameters for which the results hold (number of neurons, sparsity, global weight scaling, etc.) such that any large enough population, mixing excitatory and inhibitory neurons can quickly learn to realize such gated working memory. In a nutshell, with a minimal set of hypotheses, we show that we can have a robust model of working memory. This suggests this property could be an implicit property of any random population, that can be acquired through learning. Furthermore, considering working memory to be a physically open but functionally closed system, we give account on some counter-intuitive electrophysiological recordings.

Keywords: Working Memory, Line Attractor, Reservoir Computing, Computational Neuroscience, Prefrontal Cortex, Model

1 Supplementary

1.1 Reduced model

The effect of parameters a and b are of distinct nature as illustrated in Figure S1. b controls the time the memory takes to fade to zero whereas a controls how the model will mix the previous output and the new desired one. Interestingly, the time the information takes to fade out to zero is not directly linked to a time constant of neurons. Moreover, the mixing factor can actually be computed for b small enough and is equal to $\tanh(a)^2$. Consequently, by feeding T with a (instead of 1) and fixing the input weights coming from T to 1 (instead of a), one can obtain a model which mimics the update mechanism of a Gated Recurrent Unit (GRU) but with regular neurons.

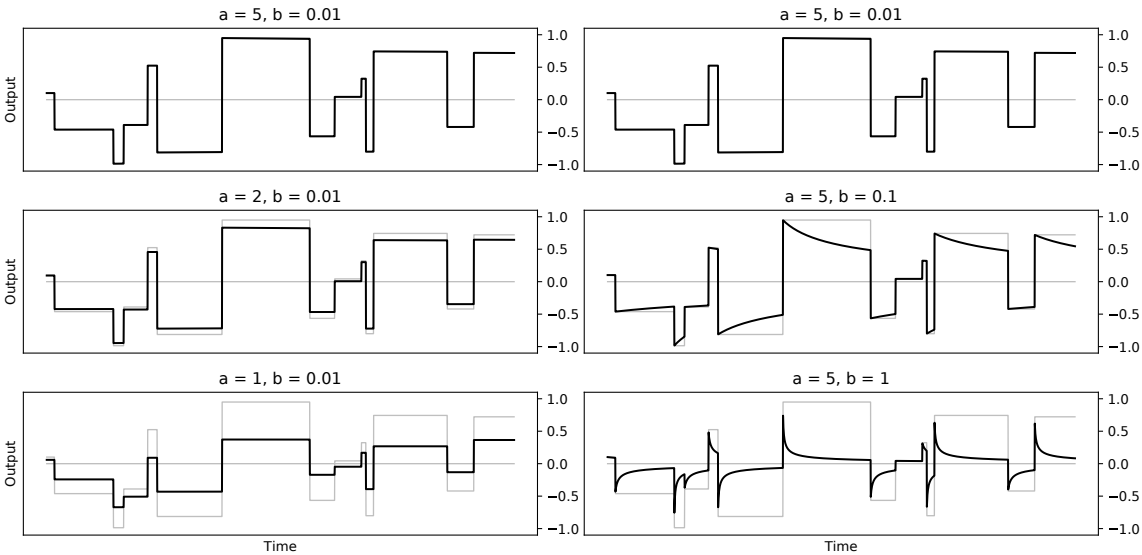


Figure S1: Influence of parameters a and b on the reduced model. The parameter b controls the amount of feedback fed to the model. If b is too high ($b = 0.1$, $b = 1.0$), the memory quickly fade to zero after a few timesteps. The parameter a controls the gate behavior and the ratio of the memorized and the new value that will constitute the new memorized value. If a is too low ($a = 1$, $a = 2$), there is a systematic undershoot.

1.2 Online learning

We tested online learning by using FORCE (Sussillo & Abbott, 2009) described by the following update equations:

$$W_{out}[n] = W_{out}[n-1] - e_{-}[n]P[n]x[n] \tag{1}$$

$$P[n] = P[n-1] - \frac{P[n-1]x[n]x^T[n]P[n-1]}{1 + x^T[n]P[n]x[n]} \tag{2}$$

where $e_{-}[n]$ represents the error made at time n if there were no update of W_{out} and $P[n]$ is a squared matrix computing an online inverse of $XX^T - \alpha I$, α a regularization term and P initialized to $P[0] = \frac{I}{\alpha}$.

In Figure S2 and S3 , one can see that the behavior obtain with online learning is similar to offline learning.

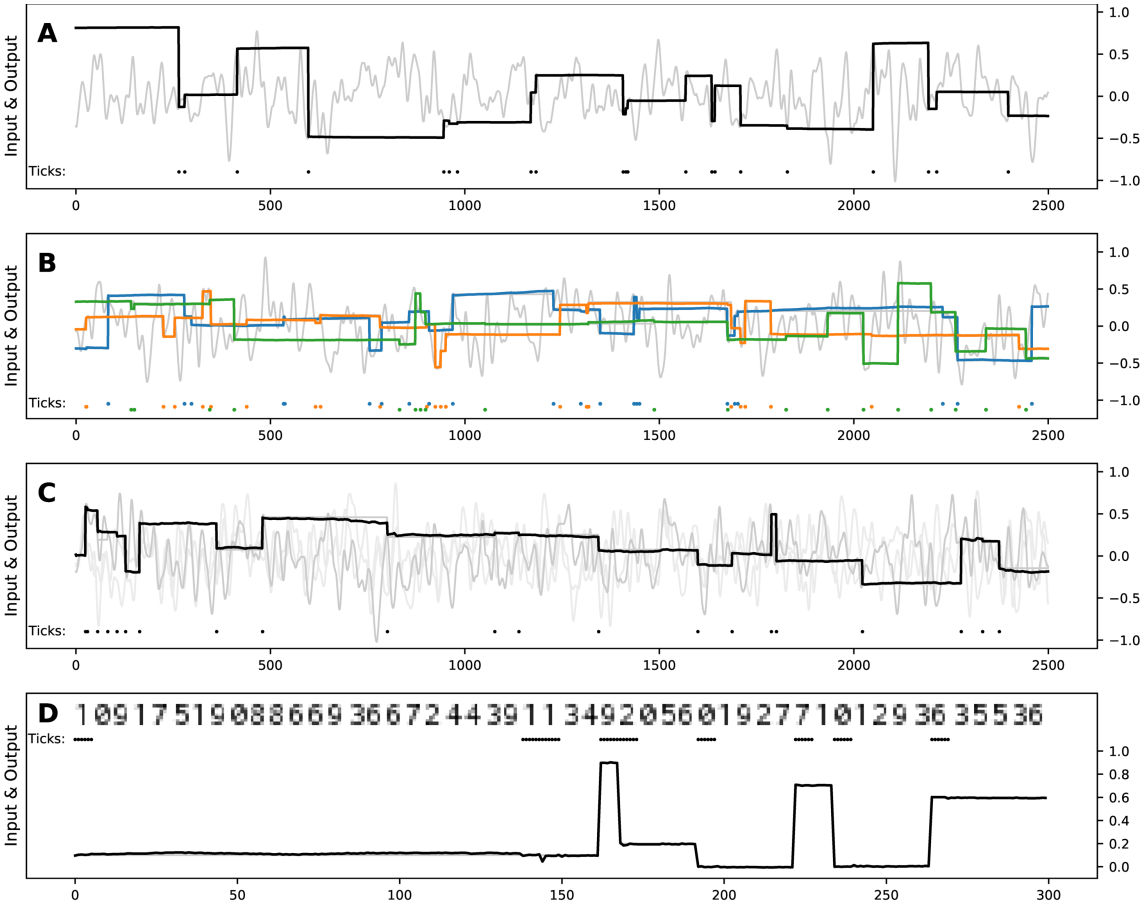


Figure S2: Performance of the reservoir model on working memory tasks with online FORCE learning. Same training/testing protocol than for Figure 9. The regularization parameter α has been fixed to 0.0001. The light gray line is the input signal and the thick black (or colored) one is the output of the model. Dots at the bottom represents the trigger signal (when to memorize a new value). For the digit task, the input containing the value to maintain is shown as a picture instead. **A** *1-value 1-gate scalar task* **B** *1-value 3-gate scalar task* **C** *3-value 1-gate scalar task* **D** *1-value 1-gate digit task.*

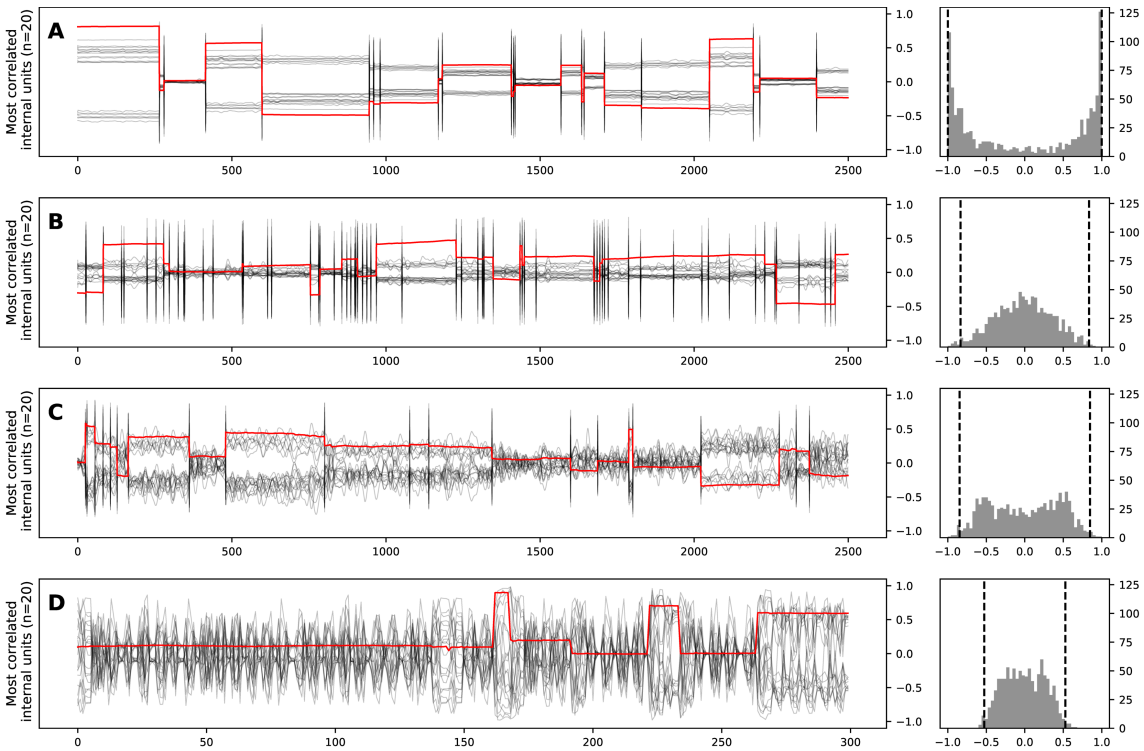


Figure S3: Most correlated reservoir units displaying various degrees of maintained activity with online FORCE learning. Same training/testing protocol than for Figure 9. The regularization parameter α has been fixed to 0.0001. (Left) in black the activities of the 20 neurons the most correlated with the output, in red the output of the model. (Right) histogram of the correlation between the neurons and the output; y-axis represents the number of neurons correlated with the output produced by the model within a correlation bin. Dashed lines indicate the 20th most correlated and most anti-correlatted neurons with the output. **A** *1-value 1-gate scalar task* **B** *1-value 3-gate scalar task* **C** *3-value 1-gate scalar task* **D** *1-value 1-gate digit task*. Most correlated reservoir units do not necessarily show clear sustained activity: we can see a degradation of sustained activities from **A** to **D** according to the different tasks. Thus, maintained activity is not compulsory to perform working memory tasks.

1.3 More on the segment attractor

In Figure S4 we performed a principal component analysis on the reservoir state obtained in Figure 8. We can note two interesting facts: (1) The activity evolves in a very low dimensional space, the first component explains more than 99% of the variance by itself, and (2) this component is linked to the memory maintained, linearly between -1 and 1 and non-linearly outside. With both combined together we can say that the segment attractor we built is actually a straight line.

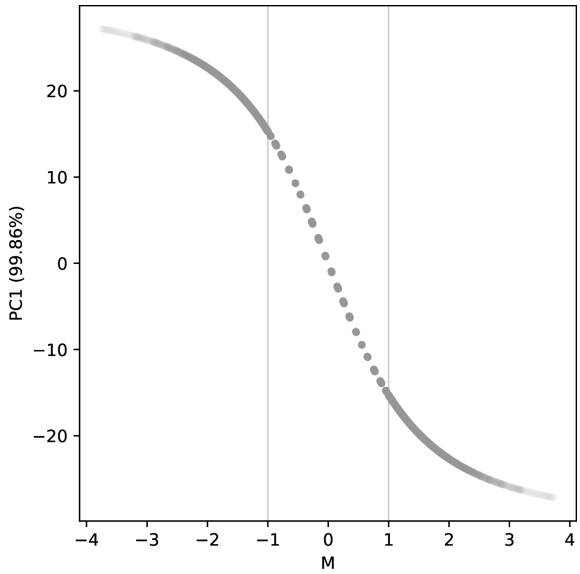


Figure S4: The line attractor. The reservoir state and output analyzed are the one of Figure 8. x-axis: output (memory). y-axis: first principal component of the reservoir state.

1.4 Influence of the spectral radius on the segment attractor

By watching how the segment attractor is evolving against the spectral radius, we can better explain its influence on the solving of the 1-value 1-gate task. The smaller the spectral radius the more precise is the approximation of the segment attractor. For a 0.1 spectral radius the approximate segment attractor seems continuous. The difference is quasi-null for starting values in [-1,+1] while for values outside this range, the difference corresponds to the difference with the nearest bound of the [-1,+1] segment. For a 0.5 spectral radius we clearly can see that the segment attractor is discretized. In the very first step the outputs concentrates around discretized value between -1 and 1 and is kept constant. In the extreme case for a 1.0 spectral radius it is discretized into two points corresponding to -1 and 1.

1.5 Killing neurons having a sustained activity

In Figure S6 we show how the neurons are correlated with the output (M), the value (V) and the trigger (T) for different feedback scaling. Because some irregularities come with triggers, we removed the trigger time steps while computing the correlation with the output and the value. We can note that the profile of neurons seems to change according to the feedback scaling. When it is high (1.0), there is mostly neurons following the output, when it is low (0.1) few neurons continue to follow

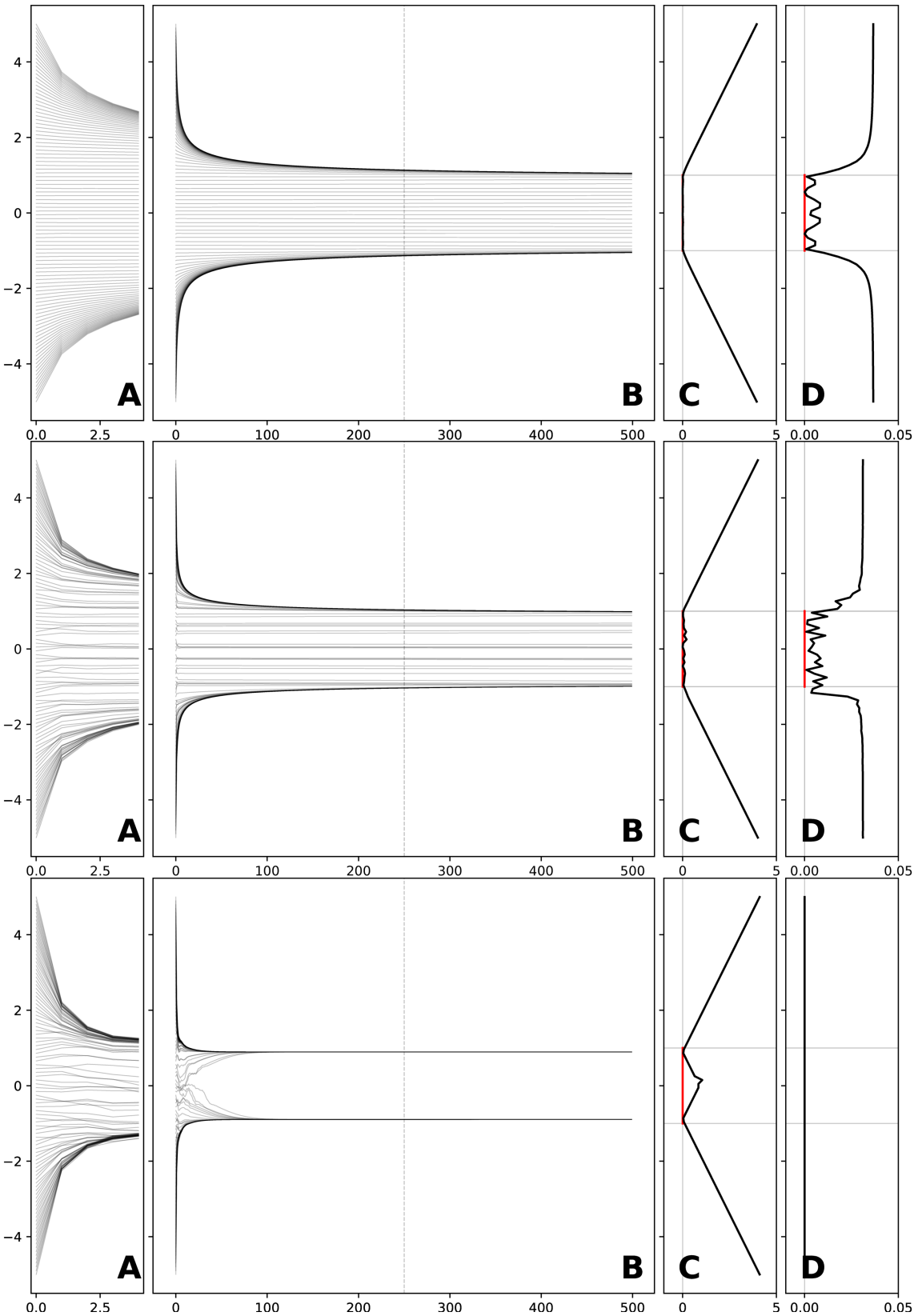


Figure S5: Influence of spectral radius on approximate line attractor The same trained models were tested for 500 iterations without inputs, only starting with an initial trigger along with values linearly distributed between -5 and +5. First line: Spectral radius 0.1. Second line: Spectral radius 0.5. Third line: Spectral radius 1.0. **A** and **B** Output trajectories. Each line corresponds to one trajectory (output value) of the model. **A** is a zoom of **B** on the first few time steps. **C** Asolute difference in the output between initial time and and final time. **D** Maximal absolute difference in the neurons between intermediate (dashed line) and final time.

the output while most of them vary according to the value they receive as input. To better identify the neurons which are following the output we looked at where were located the weights of these neurons in Figure S7. We can note that the most correlated neurons with the output are mainly the ones with relatively small input weights coming from the value. To go further with this idea we changed the sampling of the input weights to remove small values. In practice the sampling was done uniformly in [-1,-0.5]∪[0.5,1.0] instead. The behavior of this modified model is shown in Figure S8. We can note that the model is still able to perform the task relatively well, just a little worse than before, but there is no more neuron displaying a sustained activity. Interestingly, in Figure S7, the neurons the most correlated with the trigger displays an intringuing property, their output weights seems to be correlated to their feedback weights.

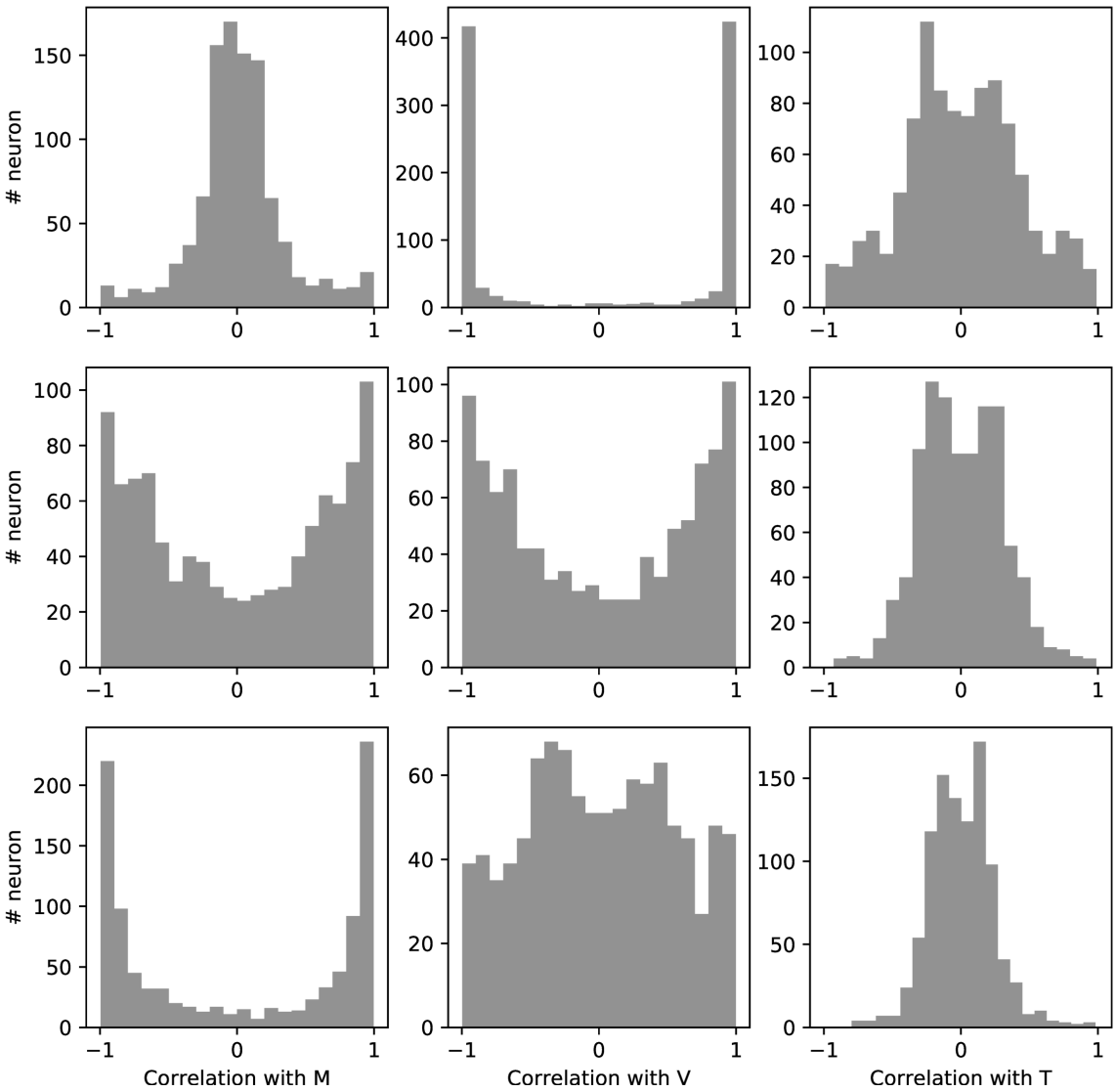


Figure S6: Influence of feedback scaling on maintenance of neurons First line: Feedback scaling 0.1. Second line: Feedback scaling 0.5. Third line: Feedback scaling 1.0. First column: Correlation with M. Second column: Correlation with V. Third column: Correlation with T.

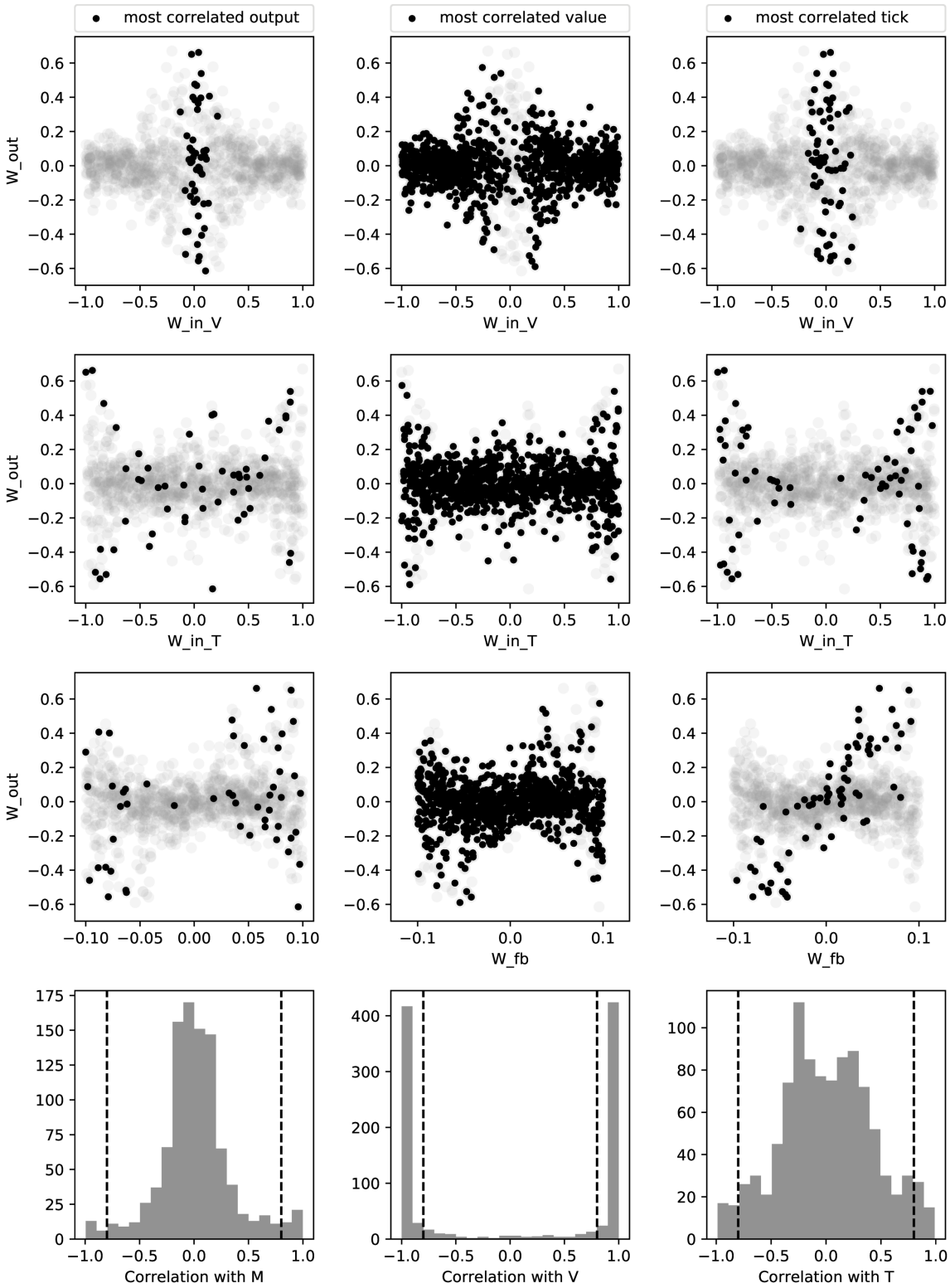


Figure S7: Link between weights and behavior of the neurons The weights of the neurons the most correlated with either M, V or T are highlighted. First column: M. Second column: V. Third column: T. The three first lines show the link between the output weights and some other weights. First line: weights coming from V. Second line: weights coming from T. Third line: weights coming from M. The fourth line shows the correlation with the quantity used to highlight. The threshold used to decide which are the most correlated is shown in dotted black.

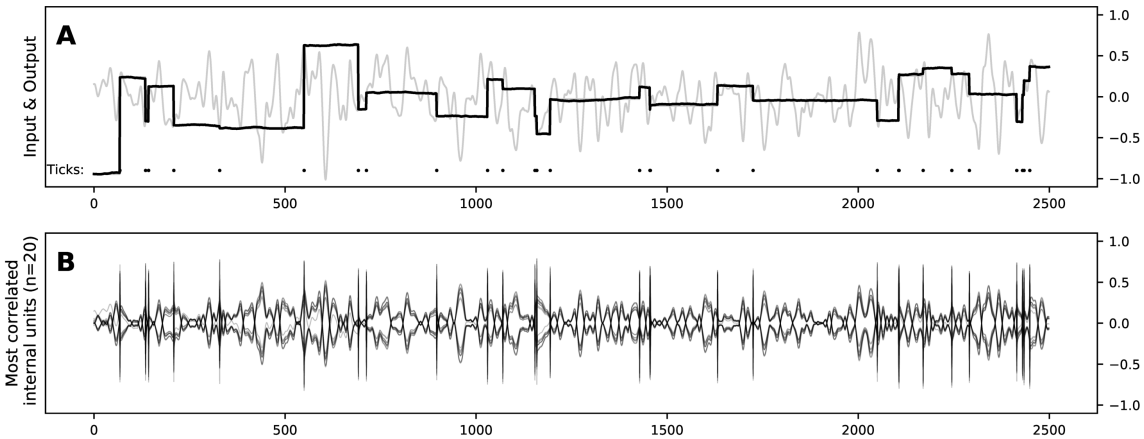
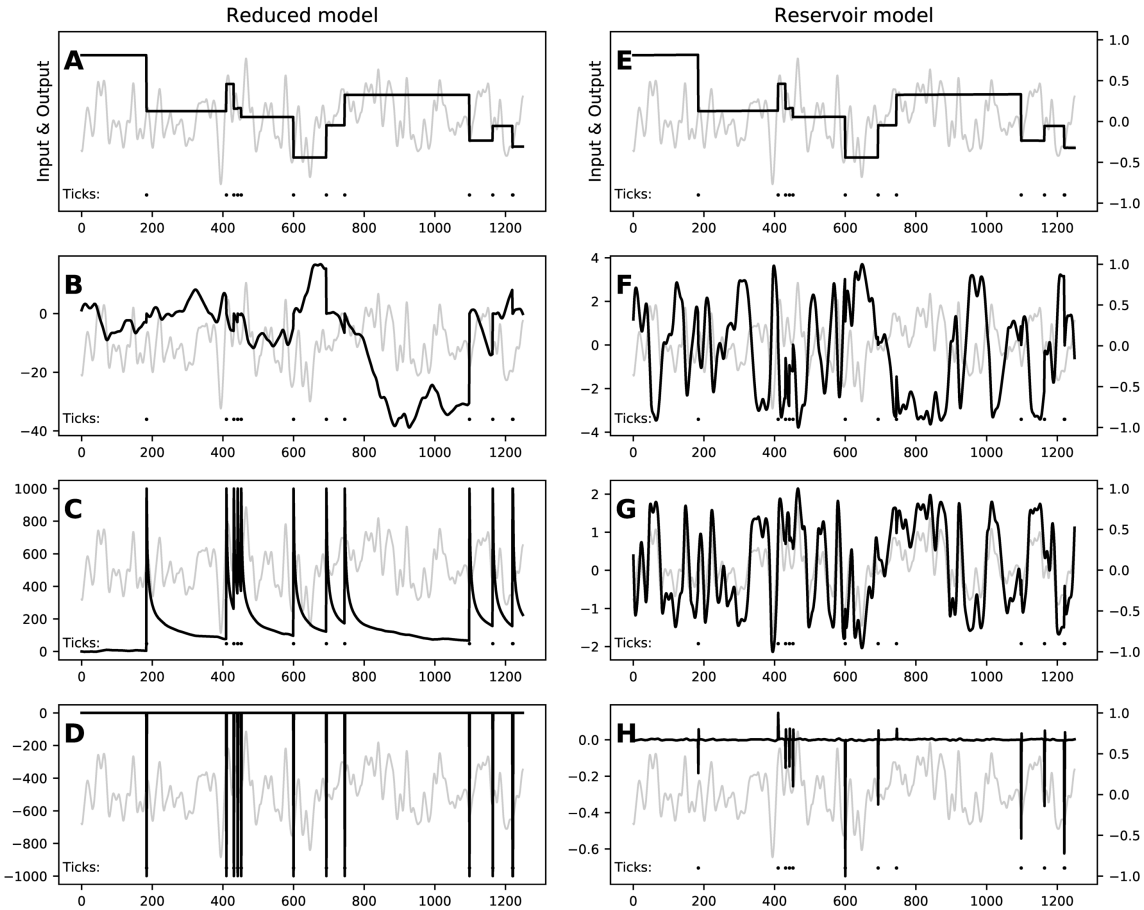


Figure S8: Performance of the model with high input value weights on the 1-gate task. A The light gray line is the input signal and the thick black one is the output of the model. Black dots at the bottom represents the trigger signal (when to memorize a new value). **B** Activity of 20 units that are the most correlated with the output.

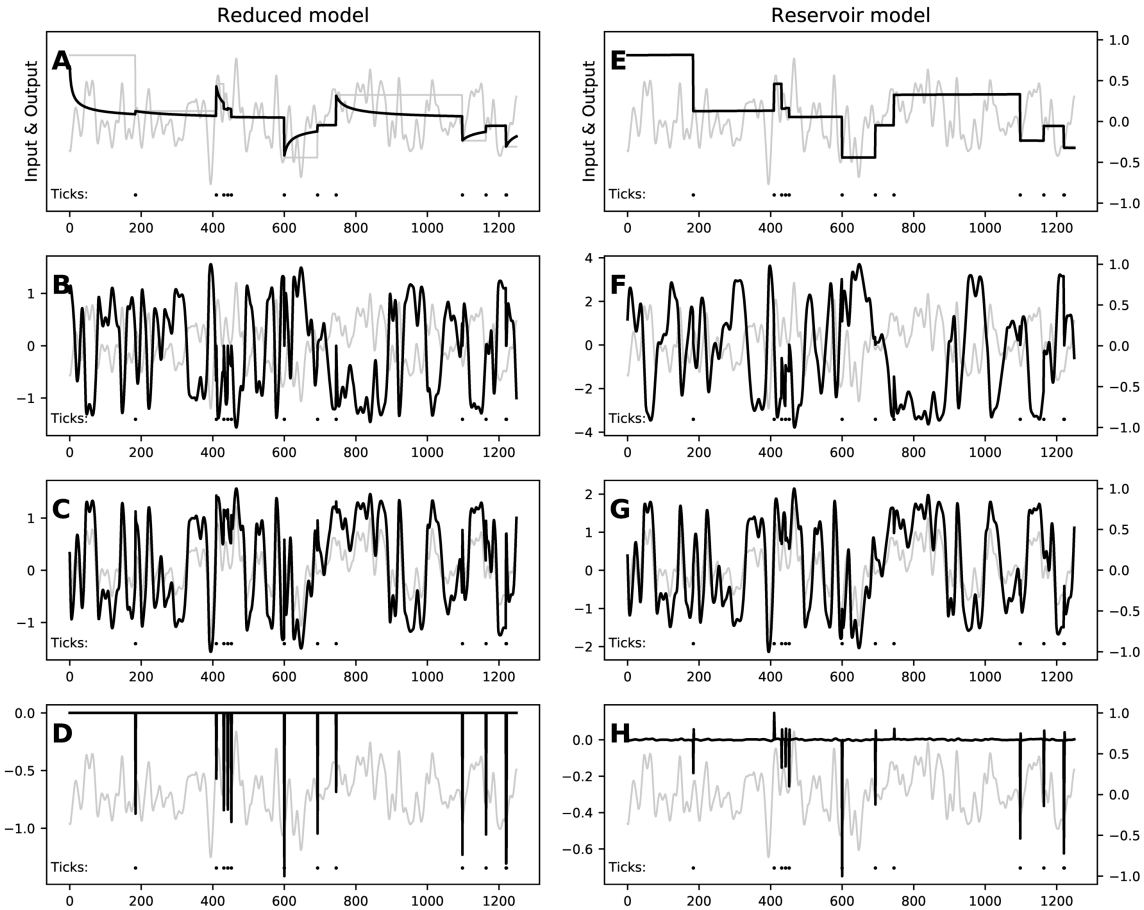
1.6 Comparative lesion studies

To explore further the link we established between the reservoir and the reduced model, we conducted comparative lesion studies by forcing a given population output to be 0 (anytime) in both models. We first compared the original reduced model ($a = 1000, b = 0.001$) with the reservoir model. However, since b is very small, and since X_2 and X_3 saturates, their contribution to the output become very large in the presence of a trigger, making the comparison irrelevant (see Figure S9). We thus considered an alternative reduced model ($a = 1000$, $b = 1$) to prevent the output to become very large. This alternative intact model is not able to sustain the working memory anymore but the output lesioned model is now strongly correlated with the output of the reservoir model (see Figure S10). This further tighten the link between the different subpopulation X_1/R_1, X_2/R_2 and X_3/R_3 even though it is only valid for this specific set of parameters.



Silenced populations	Output correlation
X1 / R1 (T=0)	0.2995
X2 / R2 (T=0/1)	0.1400
X3 / R3 (T=0/1)	0.7549

Figure S9: Side by side comparison of lesions in the minimal (left) and full (right) models. $a = 1000$, $b = 0.001$. **A & E** No lesion. **B & F** X1-R1 lesioned. **C & G** X2-R2 lesioned. **D & H** X3-R3 lesioned. Bottom table: Correlation between output of reduced and reservoir model when one of the component is silmenced. Right y-axis: Input (gray) Left y-axis: Output (black)



Silenced populations	Ouput correlation
X1 / R1 (T=0)	0.8576
X2 / R2 (T=0/1)	0.9834
X3 / R3 (T=0/1)	0.8785

Figure S10: Side by side comparison of lesions in the minimal (left) and full (right) models. $a = 1000$, $b = 1$. **A** & **E** No lesion. **B** & **F** X1-R1 lesioned. **C** & **G** X2-R2 lesioned. **D** & **H** X3-R3 lesioned. Bottom table: Correlation between output of reduced and reservoir model when one of the component is silenced. Right y-axis: Input (gray) Left y-axis: Output (black)