INFERNO: A novel architecture for generating long neuronal sequences with spikes

Alex Pitti, Philippe Gaussier, and Mathias Quoy

Laboratoire ETIS, CNRS UMR 8051, University of Cergy-Pontoise, ENSEA, France
alexandre.pitti@u-cergy.fr
http://www.etis.ensea.fr

Abstract. Human working memory is capable to generate dynamically robust and flexible neuronal sequences for action planning, problem solving and decision making. However, current neurocomputational models of working memory find hard to achieve these capabilities since intrinsic noise is difficult to stabilize over time and destroys global synchrony. As part of the principle of free-energy minimization proposed by Karl Friston, we propose a novel neural architecture to optimize the free-energy inherent to spiking recurrent neural networks to regulate their activity. We show for the first time that it is possible to stabilize iteratively the long-range control of a recurrent spiking neurons network over long sequences. We identify our architecture as the working memory composed by the Basal Ganglia and the Intra-Parietal Lobe for action selection and we make some comparisons with other networks such as deep neural networks and neural Turing machines. We name our architecture INFERNO for *Iterative Free-Energy Optimization for Recurrent Neural Network*. *abstract* environment.

Keywords: Free-Energy, Predictive Coding, Working Memory, Neuronal Sequences, Spiking neurons, STDP, Basal Ganglia, Cortico-basal Loops, Habit Learning

1 Introduction

Hierarchical plans and tree structures are a hallmark for human language and cognition [2]. But how the brain does to construct and retrieve them dynamically? For instance, the spontaneous activity within the network rapidly perturbs the neural dynamics and it is rather difficult then to maintain any stability for controlling long-range synchrony.

Making an analogy with the butterfly effect in chaos theory, small perturbations can destroy the dynamics even after few iterations. At reverse, exploiting this intrinsic noise can serve to make to converge neural dynamics to attractors, as a chaotic itinerancy [12]. In spiking neural networks, we propose that the tiny control of the neurons' sub-threshold activity (small events) can drive at another order of magnitude the generation of spikes (big events).

As a novel mechanism, we propose to exploit this intrinsic noise to regulate the neural activity and the neurons' firing; an idea in line with the free-energy

minimization principle proposed by Karl Friston [3]. The minimization of the free energy means to predict for one particular policy its expected state and to optimize it over time in order to minimize future errors [4]. Our neural model is based on this principle of *Iterative Free-Energy Optimization for Recurrent Neural Networks*, and we named it INFERNO [9], see Fig. 1.

Moreover, this architecture is supported by several proposals and observations that consider the functional organization between the cortex with the subcortical regions (the basal ganglia); c.f. [10, 6, 11, 8, 1, 7].

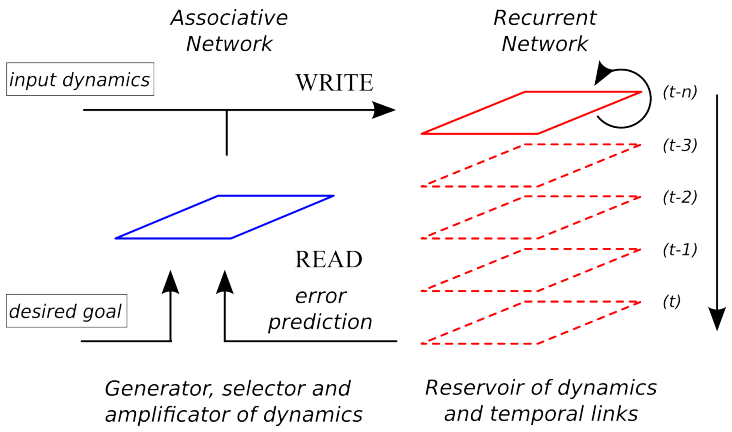


Fig. 1. Neural architecture INFERNO for Iterative Free-Energy Optimization of Recurrent Neural Networks. This neural architecture is based on the coupled system formed by an associative memory (AM) and a recurrent neural network (RNN). INFERNO generates, selects and stores a set of rules in AM to assemble dynamically a neuronal sequence from a reservoir of dynamics in RNN toward a desired goal and based on free-energy minimization. It has some similarities with a Turing machine that a table of instructions, Write and Read heads to generate a code from an infinite tape.

2 Methods

We use the Rank-Order Coding (ROC) algorithm to model spiking neurons and the learning mechanism of Spike Timing-Dependent Plasticity [13]. The neurons' output y is computed by multiplying the rank order of the sensory signal vector x, $f(x) = \frac{1}{rank(x)}$, by the synaptic weights w so that $y = \sum wf(x)$. The function $rank(x)$ corresponds to the *argsort* function in Matlab and the synaptic weights of the neurons Δw are updated at each iteration.

The first neural architecture consists of one recurrent neural network of spiking neurons RNN arranged as in Fig 1 a) in red. The second neural network consists on one associate neural network ANN as in Fig 1 a) in blue. The output

vector y of RNN is compared to one desired goal activity y^* to compute the error prediction $e = y^* - y$. Based on the variational error Δe, a stochastic descent gradient is used to generate the input vector x that will minimize error e on the long-term. The ANN learns to reconstruct the RNN input vector x based on the error prediction e on the output vector y so that $x = \sum vf(y)$. The coupled system composed of ANN and RNN attempts to minimize error dynamically. The former learns to control the latter system to generate a temporal sequence directly based on the feeded back activity. We describe below the stochastic search algorithm.

Table 1. Free-energy optimization based on stochastic gradient descent to minimize prediction error.

Stochastic optimization as Accumulation Evidences Process
#01 At time $t = 0$, initialize V, V^*, I
#02 choose randomly I_{search}
#03 compute $V_{search}(t)$ from $V(t)$, $I + I_{search}$
#04 **While** $t \leq$ *horizon_time*, repeat:
#05 compute $V_{search}(t+1)$ from $V_{search}(t)$
#06 **If** $V^* - V \geq V^* - V_{search}(t+1)$:
#07 $I = I + I_{search}$
#08 $V = V_{search}$
#09 **break**
#10 $t = t + 1$
#11 **Goto** #02

Prediction error E on the output vector V is used as a reinforcement signal to control the level of noise I_{search} to inject in the input dynamics I in order to explore local or global minima toward V^*.

3 Results

We propose at first to explain how ANN controls the neural dynamics of RNN with respect to one goal vector and error prediction relative to it. We plot in Fig 2 the dynamics of RNN after several iteration of error descent gradient and explorative search till discovery of the solution (a). After few iterations, ANN finds the input dynamics that makes to converge RNN (b). That is, the coupled system self-organizes itself to minimize error toward a goal dynamically (c-d).

We propose to use the ANN-RNN architecture for controlling one 3 DOF arm motion toward goals that we give on the fly, see Fig. 3. Only three RNN neurons control the normalized angles of the robot. We emphasize also hat ANN has no information about it, just about distance error between the location of the end

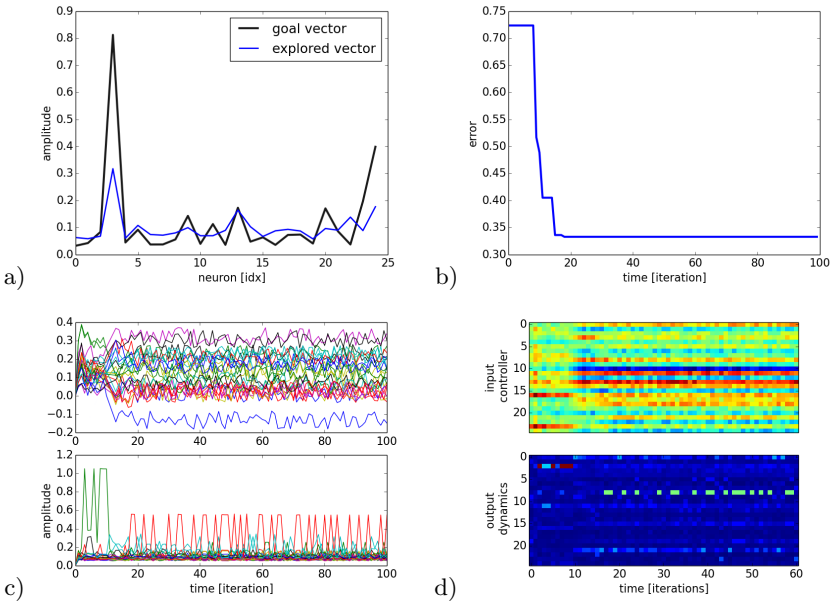


Fig. 2. Explored dynamics toward a goal. (a) ANN makes to converge the RNN neurons (in blue) to some desired dynamics (in black) thanks to prediction error. (b) the ANN prediction error diminishes to reach a local minimal value during the first 20 iterations. (c-d) Dynamics of the ANN to control the dynamics of the RNN.

effector and the location of the target. We change dynamically the target place and the arm is searching for a new configuration that minimizes error. ANN dynamics are changing everytime the target is placed at a new location (middle, upper chart), as do the three neurons of RNN, which converge to angles that reach the goal (middle, lower chart). Over time, each ANN neuron learns the dynamics that control the RNN, see Fig 3. This neural architecture is capable to generate neuronal sequences based on habit learning once ANN has learned to control RNN (exploitation), before this happens, ANN searches to minimize error in a supervised manner toward a desired goal (exploration).

We can let the two coupled networks to self-organize their dynamics so that ANN triggers one specific neural trajectory in RNN, which triggers back one ANN neuron, the most problable one from the generated neural trajectory, see Fig 4. For each neuron in ANN, ANN triggers one specific rule to direct RNN dynamics. The most probable ANN rule is then selected depending on the y vector found in RNN. In this way, the two systems control themselves to generate serial neuronal sequence several for hundreds of iterations without error. We emphasize that these dynamics are not completely learned but generated off-the-shelf.

In an experience of serial ordering computation, ANN learns simple connecting rules to trigger the RNN dynamics. During this task, eventhough error and

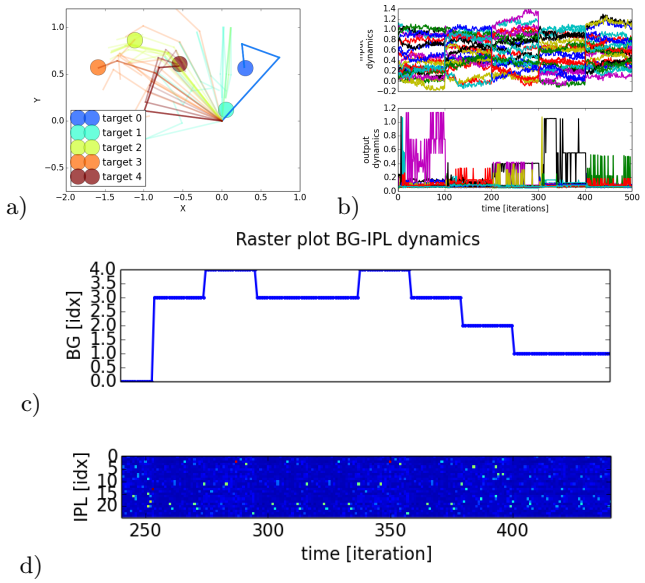


Fig. 3. RNN Arm control based on predictive coding. On the left, RNN controls the three joint angles of one planar robot. ANN controls RNN based on error prediction toward targets. In the middle, ANN and RNN dynamics when switching dynamically to the euclidean distance to the goal location furnishes a reward to the motor neurons. On the right, ANN controls the neural activity of RNN over time.

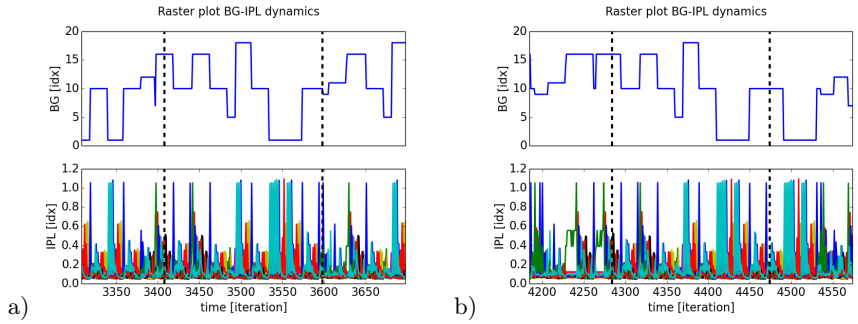


Fig. 4. ANN-RNN Long-range neuronal sequences. Each neuron of ANN can control the neural dynamics of RNN for a relative long period (20 iterations in our case), which in return selects the correct ANN neuron to pursue the sequence. By doing so, the coupled system can produce the serial ordering of chunks toward multistep computation. The same sequence is generated within the dashed lines.

variability occur they are minimized dynamically to retrieve the goal vector. The three trajectories of spiking neurons present similar dynamics, mixing variability and robustness. The coupled system ANN-RNN formed by INFERNO presents

some of the properties of a working memory to be robust and flexible at the same time. To some point, INFERNO appears to overcome the exploration and exploitation dilemna of machine learning algorithms thanks to predictive coding.

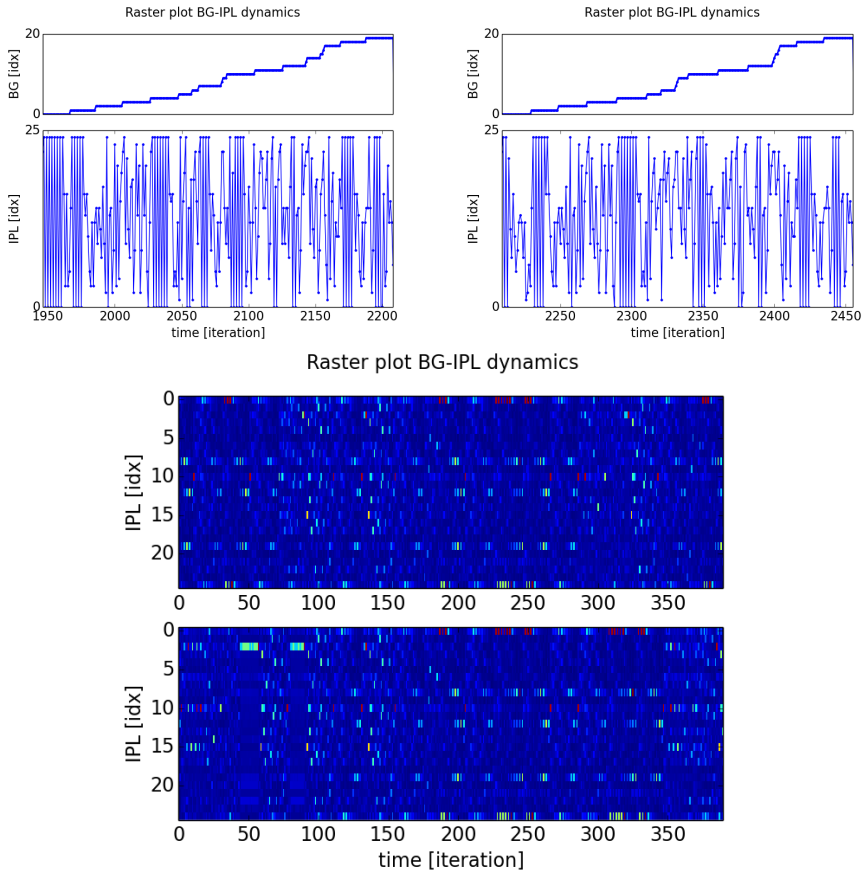


Fig. 5. Self-Organized serial ordering sequence. Three examples of dynamic sequence ordering show that self-organization is not rigid and that variability occurs during time. The error minimization serves to rebind the two systems from each other.

4 Discussion

We propose a framework based on a coupled recurrent spiking neuronal system that achieves to perform long sequential planning by controlling the amplitude level of the spiking neurons through reinforcement signals. The control done is weak so that the propagated reinforced signals let the working memory plastic

enough to converge to the desired internal states from various trajectories. Used in a robotic simulation, the neural dynamics can drive a three d.o.f. arm to reach online different locations.

The neural control is done by controlling tiny variations injected into the recurrent network that can iteratively change its dynamics to make it to converge to attractors. To this respect, our framework embodies some aspects of the free-energy optimization principle proposed by [4].

INFERNO generates, selects and stores a set of rules to assemble dynamically a neuronal sequence from a reservoir of dynamics toward a desired goal and based on free-energy minimization.

While the RNN working memory provides, stores, and manipulates representations; the ANN maps current states to courses of action. ANN can serve for selection of complex, sequenced actions at RNN. Thus, it can be interpreted as a repertoire of if-then rules or a set of stimulus-response associations to select appropriate cortical chains. To some points, we think it has some similarities with a Turing machine with a table of instructions, Write and Read heads to generate a code from an infinite tape [14] [5]. Unwrapped in time, INFERNO generates tree-like trajectories as a A* algorithm and as a virtually deep feed-forward neural network, see Fi. 6. With INFERNO, we make a parallel with the cortico-basal system to construct a working memory. Iteratively, the basal ganglia forms 'habits' or rules that select cortical primitives in order to generate neuronal sequences based on a desired goal provided by the prefrontal cortex. The reinforcement signals given by the dopaminergic neuromodulator is similar to error prediction optimization or to free-energy minimization.

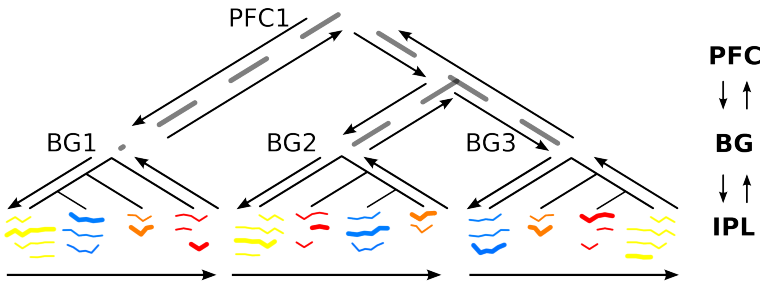


Fig. 6. Working memory for tree-like sequences. Unwrapped in time, INFERNO generates tree-like trajectories as a A* algorithm and as a virtually deep feed-forward neural network. RNN has neuronal primitives that ANN can selects, amplifies. This is similar to cortico-basal loops, having the basal ganglia to control the dynamics of the cortical maps and learns context-dependent rules depending on prefrontal cortex goal state.

Acknowledgments. This work was partially funded by EQUIPEX-ROBOTEX (CNRS), chaire dexcellence CNRS-UCP and project Labex MME-DII (ANR11-LBX-0023-01).

References

1. Benedek, M., Jauk, E., Beaty, R., Fink, A., Koschutnig, K., Neubauer, A.: Brain mechanisms associated with internally directed attention and self-generated though. Scientific Reports 6, 22959 (2016)
2. Dehaene, S., Meyniel, F., Wacongne, C., Wang, L., Pallier, C.: The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. Neuron 88, 2–19 (2015)
3. Friston, K.: A theory of cortical responses. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 360(1456), 81536 (2005)
4. Friston, K., Kilner, J.: A free energy principle for the brain. J. Physiol. Paris 100, 70–87 (2006)
5. Graves, A.e.a.: Hybrid Computing Using a Neural Network with Dynamic External Memory. Nature 538, 471476 (2016)
6. Guthrie, M., Leblois, A., Garenne, A., Boraud, T.: Interaction between cognitive and motor cortico-basal ganglia loops during decision making: a computational study. Journal of Neurophysiology 109, :3025–3040 (2013)
7. Koechlin, E.: Prefrontal executive function and adaptive behavior in complex environments. Current Opinion in Neurobiology 37, 1–6 (2016)
8. Miller, E.: The "working" of working memory. Dialogues Clin Neurosci. 15(4), 411–418 (2015)
9. Pitti, A., Gaussier, P., Quoy, M.: Inferno: Iterative free-energy optimization for recurrent neural networks. PLoS One p. to appear (2017)
10. Seger, C., Miller, E.: Category learning in the brain. Annu. Rev. Neurosci. 33, 20319 (2010)
11. Topalidou, M., Rougier, N.: [re] interaction between cognitive and motor cortico-basal ganglia loops during decision making: a computational study. ReScience (2015)
12. Tsuda, I., Fujii, H., Tadokoro, S., Yasuoka, T., Yamaguti, Y.: Chaotic itinerancy as a mechanism of irregular changes between synchronization and desynchronization in a neural network. J. of Integr. Neurosc. 3, 159–182 (2004)
13. Van Rullen, R., Gautrais, J., Delorme, A., Thorpe, S.: Face processing using one spike per neurone. BioSystems 48, 229–239 (1998)
14. Zylberberg, A., Dehaene, S., Roelfsema, P., Sigman, M.: The human turing machine: a neural framework for mental programs. Trends in Cognitive Sciences (7), 293–300 (2011)