Brain-like features of MemComputing machines

Massimiliano Di Ventra, *Fellow, IEEE*
Department of Physics
University of California, San Diego
La Jolla, CA 92093, USA
diventra@physics.ucsd.edu

***Abstract*—MemComputing is a new model of computation that exploits the non-equilibrium property—we call "memory"—of any physical system to respond to external perturbations by keeping track of how it has reacted at previous times. Its digital, scalable version maps a finite string of symbols into a finite string of symbols. In this paper, I will discuss some analogies of the operation of MemComputing machines—in general, and digital in particular—with a few physical properties of the biological brain. These analogies could be a source of inspiration to improve on the design of these machines. In turn, they could suggest new directions of study in (computational) neuroscience.**

***Index Terms*—MemComputing, brain, neurons, instantons**

I. Introduction

Although the complete understanding of how the biological brain works is still far from complete, some of its features have been experimentally determined. These include, for instance, the main components and operation of a single neuron, the establishment of short- and long-term memories [1], the long-range correlations in the dynamics of clusters of neurons [2], etc. Neuro-inspired computation, when implemented in solid-state hardware [3] or emulated in artificial neural networks [4], aims at reproducing and/or exploiting some of these features for some tasks such as learning, association, pattern recognition, prediction, and so on.

Recently, a new model of computation has been introduced, named *MemComputing* [5]–[7], which takes advantage of a physical property shared by all physical systems: *memory* (another name for "time non-locality"). This *non-equilibrium* property means that when the state of a physical system is perturbed, the perturbation affects the system's state at a later time [7], [8]. Of course, in some cases the memory decays so fast that it is not experimentally detectable, or it is too small to be technologically useful. However, there are situations in which time non-locality is strong enough that it can be exploited for computing and that is the main idea behind MemComputing [7].

In this paper I will discuss some analogies MemComputing machines (in general, and digital in particular) share with the operation of the biological brain. Some of these analogies are *emergent phenomena* of the dynamics of these machines. Therefore, they could shed further light on the operation of the biological brain itself.

II. Universal MemComputing machine

Let us start from the general definition of the *universal* MemComputing machine (UMM) [6].[1] A UMM is defined as the eight-tuple [6]

$$UMM = (M, \Delta, \mathcal{P}, S, \Sigma, p_0, s_0, F)\,. \tag{1}$$

Here, M indicates the set of possible states of a single memprocessor (the fundamental unit of a MemComputing machine). The set $\mathcal{P}$ contains the arrays of pointers, p_α, that select the memprocessors called by the *transition function* δ_α. The set of indexes is indicated with S. Σ is the set of initial states written by the input device on the computational memory. $p_0 \in \mathcal{P}$ is the initial array of pointers, s_0 is the initial index α, and $F \subseteq M$ is the set of final states.

The set of *transition functions*, Δ, has elements

$$\delta_\alpha : M^{m_\alpha} \backslash F \times \mathcal{P} \to M^{m'_\alpha} \times \mathcal{P}^2 \times S\,, \tag{2}$$

with a number $m_\alpha < \infty$ of input memprocessors (read by the transition function δ_α), and $m'_\alpha < \infty$ output memprocessors written by the same transition function.

I note first that it was shown in [6] that the mathematical definition of the UMM encompasses also the description of artificial neural networks (ANNs). In other words, ANNs can be viewed as a *special case* of MemComputing machines. However, Eqs. (1) and (2) allow us to go beyond this realization and show additional similarities with the operation of the biological brain. These are as follows.

A. Massively parallel architecture with combined information processing and storage

Any transition function δ_α of a UMM simultaneously acts on a set of memprocessors at once. This was named "intrinsic parallelism" in [6] and it is fundamentally different from the "standard parallelism" of our modern computers (or even parallel Turing machines). Instead, this feature seems to belong to the biological brain, or at the very least, its representation as an artificial neural network [9].

In addition, by construction, memprocessors and their network (*computational memory*) can process and store information simultaneously. Although it is still not fully clear how the brain performs these two tasks, compelling evidence points to the collection of neurons and synapses in the brain as the main agents able to concomitantly carry out these functions [1].

This work was supported by the National Science Foundation under Grant ECCS-2229880.

[1] The term "universal" means that this machine is Turing-complete [6].

B. Asynchronous computation

Asynchronous computation means that all or a large chunk of processing units of a machine compute and exchange information simultaneously, without the need to wait for a predetermined period of time, such as the global clock period in our modern computers. Asynchronous computation is a main feature of the biological brain, and models of artificial neural networks. It also follows from the general definition of MemComputing machines; cf. Eqs. (1) and (2). These machines do not require a global clock, and the different memprocessors compute and exchange information simultaneously.

C. Information overhead

From the definition of a UMM as a collection of memprocessors it is clear that the *topology* of such a network needs to be specified and represents a fundamental aspect of these machines. This property has been called *information overhead* [6] and it is a type of “data compression” which is embedded in the machine at the outset of the computation, and does not vary during dynamics. Different types of information overheads (topologies) can be assumed ranging from polynomial to exponential [6]. It is indeed this property that allows a UMM with exponential information overhead to solve NP-complete problems with polynomial resources [6], [7]. (Note that this statement does *not* imply that NP=P, since MemComputing machines are *not* Turing machines.)

A similar concept has been discussed in the context of the brain [9]. In fact, the brain appears to have a high level of specificity, in both the types of neurons and their network architecture (the connectome), even at a mesoscopic level. This indicates that the brain physical topology is not completely random. Rather, it shows some degree of specialization (information overhead) with important functional properties [1].

D. Functional polymorphism

The set Δ of transition functions (2) of a UMM may contain more than a single element. This means that a UMM can, in principle, compute *different functions* without modifying the topology of the machine network, by simply applying the appropriate input signals. This feature was named *functional polymorphism* [6], and it is not available to our modern computers (or Turing machines). A practical realization of this concept was reported in [10]. The biological brain does have this property to a certain degree. In fact, the brain can perform a wide variety of tasks without changing substantially the physical topology of its network, by simply responding to external stimuli [9].

E. Analog vs. digital computation

MemComputing machines can be defined as both analog and digital (or mixed) according to the structure of the set M of possible states of a single memprocessor. If that set is finite then the machine is digital. If it is a continuum or infinite discrete set of states then the machine operates in the analog regime. Finally, if it is some direct sum of the previous two types of sets, then the machine operates in a mixed digital-analog regime. Of course, only the digital MemComputing machines (DMMs) are easily scalable. The brain, instead, seems to operate mainly in the analog regime [1]. However, it is interesting to note that also DMMs showcase features similar to the elementary building blocks of the brain (the neurons) and their collective behavior. In the next Section, I will expand on this analogy.

III. Digital MemComputing machines (DMMs)

As already mentioned, if the set M of possible states of a single memprocessor is finite, then the machine is digital. The next question is then whether such a machine can be realized in hardware. In order to answer this question, a new set of gates, called *self-organizing gates* (SOGs) have been introduced [11]. These are terminal agnostic gates able to always satisfy their logical proposition irrespective of whether the incoming signals are from the traditional input or the traditional output. The key for their realization is the coupling of the variables of the problem—DMMs are designed to solve—with memory degrees of freedom.

A. Short- and long-term memories

To make this discussion more concrete, I report here the dynamical equations representing a DMM designed to solve for the ground state of a spin glass model Hamiltonian:

$$H = -\sum_{i>j} J_{ij} s_i s_j, \quad s_i \in \{-1, 1\}, \tag{3}$$

where the interaction strength J_{ij} (between the spin variables s_i placed on a the sites of a lattice) is random, and may involve only nearest neighbor spins or any type of interaction between spins, e.g., long range.

To design a DMM that solves such a problem, the spins are first linearized (namely they acquire a continuous value between -1 and $+1$), and then they are coupled to two types of memory degrees of freedom (short- and long-term memory) so that the phase space of the spin plus memory dynamics has only saddle points and equilibria representing the solution of the problem [12]. The full set of equations is then:

$$\begin{aligned} \dot{s}_i &= \alpha \sum_j J_{ij} s_j - 2\beta \sum_j x^s_{ij} s_i, \\ \dot{x}^s_{ij} &= \gamma C_{ij} - x^l_{ij},\ x^s_{ij} \in [0,1] \rightarrow \textit{short-term memory}, \\ \dot{x}^l_{ij} &= \delta x^s_{ij} - \zeta,\ x^l_{ij} \in [1, L] \rightarrow \textit{long-term memory}, \end{aligned} \tag{4}$$

where $C_{ij} = \frac{1}{2}(J_{ij} s_i s_j + 1) \in [0, 1]$, is a clause function, and $\alpha, \beta, \gamma, \delta, \zeta$ are time-scale parameters, fixed for all system sizes, and L is an arbitrary but finite upper limit for the long-term memory (see [12] for the choice of these parameters and for a thorough explanation of how these equation have been derived). Note that Eqs. (4) can be compactly written as $\dot{\mathbf{x}}(t) = F(\mathbf{x}(t))$, with $\mathbf{x}$ the collection of all variables, and F the flow vector field (the right hand side of Eqs. (4)).

The important point to make is that the short-term memory contains information on the recent history of the system dynamics, while the long-term memory contains information on the entire history. The existence and coupling of these two types of memories is an important ingredient to realize in practice DMMs for the solution of combinatorial optimization problems [7].

It is interesting to note that the human brain showcases also two types of memories: long-term and short-term memory [1]. The short-term memory (which is believed to be mainly located in the prefrontal cortex) is assumed to be a "working memory", namely it allows us to accomplish certain tasks that may be forgotten within a relatively short time, without much detriment. Instead, the long-term memory (which is located in the hippocampus, and from there, it is supposedly distributed to the cerebral cortex) is responsible for the storing of events far in the past. It is presumably created by the reinforcement of short memories. This is similar to how the DMMs (e.g., practically realized in Eqs. (4)) operate: the long-term memory is "reinforced" (via physical coupling) by the short-term one.

B. Instantons, action potentials, critical points and nodes of Ranvier

DMMs find the solution of a given problem by traversing specific trajectories in phase space, known as instantons [14], [15]. Instantons in DMMs are families of trajectories (a manifold) connecting a critical point (a point, $\mathbf{x}$, where the flow vector field, F, is zero) in phase space with another more stable critical point [7]. Instantons are sudden and relatively short bursts (avalanches) of the variables around the ground potential of the system.

Once the first instanton is initiated, it propagates the excitation to the next instanton, and so forth until the system reaches an equilibrium (if it exists). The critical points in between two successive instantons act as some sort of "regenerative" centers (repeaters) of the signal that propagates in the phase space, since at a critical point the system spends enough time to "decide" on the next instanton (trajectory) to take.

The mechanism I just described is very much reminiscent of how action potentials (electrical polarization signals) propagate in myelinated axons. Action potentials, like instantons, quickly "rise and fall" around the resting potential state of the axon membrane. However, in myelinated axons they propagate in a saltatory fashion from a node of Ranvier to the next [1].

The nodes of Ranvier are myelin-sheath gaps along the axon where exchange of ions between the axon membrane and the environment can occur, so that the next action potential can be generated and travel along the myelinated part of the axon. This way the action potential can "jump" from one node of Ranvier to the next, allowing for a faster conduction of the signal. The critical points in the phase space of DMMs are then the equivalent of the nodes of Ranvier in myelinated axons.

C. Long-range order

An interesting physical property of the animal brain is the observed *scale-free* behavior in the firing of neurons, even in the absence of external stimuli [2]. This is an *emergent* property of the collection of neurons, and it has been demonstrated in several experiments, although there is still much debate regarding its origin [2].

For instance, experiments have shown that cortical neurons, when deposited on a grid of electrodes, fire collectively, and the size, S, of the neuronal avalanches (how many neurons fire together) follows a power-law distribution, $S^{-\tau}$, with τ close to $3/2$.

This is similar to the critical Borel distribution of the size of the variable avalanches (instantons) found in DMMs solving combinatorial optimization problems [13]. In fact, by means of a mean-field theory it was shown that the distribution of the size of the avalanches in DMMS is also a power-law $S^{-3/2}$, which is confirmed by numerical experiments. Together with the analogy between instantons and action potentials, this emergent property of both DMMs and the brain makes the former an interesting test bed to explore phenomena that could have implications on the latter.

D. Robustness against noise vs. fault tolerance

Since DMMs employ objects of topological nature to compute (instantons) they are robust against small perturbations and noise whose strength is not enough to affect the topological structure of the phase space. However, if the architecture of the network of memprocessors is changed (by, e.g., changing even a single SOG in the circuit), the DMM would *not* solve the original combinatorial optimization problem it was designed for. Rather, it would attempt to solve this "new" problem. In other words, while DMMs are robust against noise and small perturbations, they are not fault tolerant.

The brain instead seems to have both properties. In fact, it is known that neurons in the brain both die off and are generated continuously [1]. It is then obvious that the architecture (physical topology) of the network of neurons is not fixed in time. Despite these changes (provided they are not substantial), the brain continues to function as expected, namely it is both robust against noise/perturbations (the firing of single neurons in the brain still occurs) as well as topological changes: it is fault tolerant to a high degree.

This fault tolerance could be due to the presence of time non-locality (memory) in synapses and the fact that the brain, unlike a DMM, is not attempting to solve a specific combinatorial optimization problem whose Boolean (or algebraic) expression is well defined. Time non-locality is a feature that allows "re-routing" of information "on the fly", despite failure of single units, as it was demonstrated in networks of resistive memories [16]. That particular network was designed for the solution of the shortest-path problem. Memory (time non-locality) would still promote self organization of the network into the shortest possible path or paths, in the presence of defects in the network (created by eliminating some resistive memories).

All these results seem to suggest that the more specialized the physical architecture of a network is (as for DMMs designed for specific combinatorial optimization problems), the less robust it is to topological changes. Indeed, our brains can tackle a wide variety of tasks, but are not particularly good at solving, e.g., combinatorial optimization problems.

This fact may also be related to recent research on autism spectrum disorder (ASD). For instance, some experimental studies, employing neuro-imaging techniques, have shown structural differences in several brain regions in people with ASD compared with individuals without ASD [17]. These differences may be the reason children with ASD may sometimes show some mathematical skills outperforming non-autistic peers, while struggling in some other tasks. It may very well be that the brain structure of people with ASD is topologically more constrained than that of the general population.

IV. Conclusions

In summary, I have briefly outlined the brain-like features of MemComputing machines. Of course, these similarities do not imply that MemComputing machines are brain-like. They simply indicate that some of their dynamical properties are also observed in the operation of the biological brain.

Of particular note is that some of these properties are *emerging phenonema*, namely they emerge from the collective dynamics of the units making up these machines. For instance, long-range order in DMMs—arguably the most important property for the solution of hard combinatorial optimization problems—is a feature that originates from the time non-local response of their memprocessors [7]. In fact, DMMs enter this long-range ordered state "naturally" without tuning any parameter during dynamics. This realization may help understand the supposed critical dynamics of the brain, which is still not fully understood. Work along this direction could then be beneficial in the field of (computational) neuroscience.

References

[1] L.R. Squire, F.E. Bloom, N.C. Spitzer, S. du Lac, A. Ghosh, and D. Berg, *Fundamental Neuroscience: 3rd Edition*. Academic Press, 2008.

[2] M.A. Muñoz, "Colloquium: Criticality and dynamical scaling in living systems," Reviews of Modern Physics, vol. 90, 2018, pp. 031001.

[3] C. Meade, *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, 2016.

[5] M. Di Ventra and Y. V. Pershin, "The parallel approach," Nature Physics, vol. 9, 2013, pp. 200.

[6] F.L. Traversa and M. Di Ventra "Universal MemComputing Machines," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, 2015, pp. 2702.

[7] M. Di Ventra, *MemComputing: Fundamentals and Applications*, Oxford University Press, Oxford, UK, 2022.

[8] M. Di Ventra and Y.V. Pershin, *Memristors and Memelements: Mathematics, Physics, and Fiction*, Springer, 2023.

[9] T. Kohonen, T. *Self-organization and Associative Memory (3rd edn)*, Springer-Verlag, New York (1989).

[10] F.L. Traversa, F. Bonani, Y.V. Pershin, and M. Di Ventra, "Dynamic computing random access memory," Nanotechnology, 25,2014, pp. 285201.

[11] F. L. Traversa and M. Di Ventra, "Polynomial-time solution of prime factorization and NP-complete problems with digital MemComputing machines," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 27, 2017, pp. 023107.

[12] Y.R. Pei and M. Di Ventra "Non-equilibrium criticality and efficient exploration of glassy landscapes with memory dynamics," Physica A: Statistical Mechanics and its Applications vol. 591, 2022, pp. 126727.

[13] S. R. B. Bearden, F. Sheldon, and M. Di Ventra, "Critical branching processes in digital MemComputing machines," Europhysics Letters, vol. 127, 2019, pp. 30005.

[14] M. Di Ventra, F. L. Traversa, and I. V. Ovchinnikov, "Topological field theory and computing with instantons," Annalen der Physik (Berlin), vol. 529, 2017, pp. 1700123.

[15] M. Di Ventra and I. V. Ovchinnikov, "Digital MemComputing: from logic to dynamics to topology," Annals of Physics, vol. 409, 2019, 167935.

[16] Y.V. Pershin and M. Di Ventra, M. "Self-organization and solution of shortest-path optimization problems with memristive networks," Physical Review E, vol. 88, 2013, pp. 013305.

[17] T. Iuculano, M. Rosenberg-Lee, K. Supekar, C.J. Lynch, A. Khouzam, J. Phillips, L.Q. Uddin, V. and Menon, "Brain organization underlying superior mathematical abilities in children with autism," Biological Phsychiatry, vol. 75, 2014, pp. 223.