Purdue University | Purdue Quantum Science and Engineering Institute
DISCOVERY PARK

"Representing Probabilities as Sets instead of Numbers Allows Classical Realization of Quantum Computing"

Monday, February 24, 2020 (3:30-4:30 p.m. BRK 1001)

Gerard (Rod) Rinkus, Research Scientist,
Center for Brain-Inspired Computing (C-BRIC), Purdue, ECE

Abstract

The state of an N-qubit quantum computer (QC) can be described as a superposition of 2^N terms (eq. 1), each giving the probability [expressed as a complex *number*, α_i, called a *probability amplitude* (PA)] of the respective basis state, s_i, being observed.

$$\left|\psi\right\rangle = \alpha_1\left|s_1\right\rangle + \alpha_2\left|s_2\right\rangle + \alpha_3\left|s_3\right\rangle + \ldots + \alpha_{2^N}\left|s_{2^N}\right\rangle : s_i \in \{0,1\}^N \tag{1}$$

When implementing quantum computation on classical hardware, one might represent the basis states (and thus, the PAs) *localistically*, i.e., each s_i has its own memory location in which its PA is stored. But, this requires exponential memory. Furthermore, any atomic computational operation performed on any one PA, e.g., updating its value, performs no computational work on any of the other PAs (since they are at different physical locations), implying that exponentially many operations are needed to update the entire probability distribution over all states.

Suppose instead that we represent basis states as *sparse distributed codes* (SDCs), i.e., *sets* of binary units chosen sparsely (arbitrarily widely dispersed) in a much larger field of units, and such that the sets may intersect. Thus, in stark contrast to the localist case, all basis states are stored in physical, and purely classical, superposition. Furthermore, since any given binary unit will generally be included in the SDCs of many basis states, any computational operation performed on that unit will simultaneously affect/update all of those basis states. This is called *algorithmic parallelism*, which is an orthogonal resource to machine parallelism. The specific field we'll consider is organized as Q winner-take-all (WTA) modules, each comprised of K binary units. Thus, each basis state, s_i, will be represented by an SDC, $\phi(s_i)$, or just, ϕ_i, a *set* of Q units, one per module, and the codespace (number of unique basis states representable) is K^Q. I will describe an unsupervised learning model in which this field is connected to an input field and recurrently to itself via complete (all-to-all) binary weight matrices, all weights initially zero, and SDCs are assigned to inputs (i.e., stored) as they are presented (i.e., on-line learning). Crucially, there is a purely classical single-trial learning algorithm that: a) statistically (approximately) preserves similarity, i.e., assigns more similar inputs to more similar, i.e., more highly intersecting, SDCs; and b) runs in "fixed time", i.e., the number of operations needed to store an item remains constant as the number of stored items grows. And, because the inputs are stored in a similarity-respecting way, best-match retrieval can also be done in fixed time.

Because SDCs are chosen in similarity-respecting fashion and are all the same size (Q), it follows that when any one SDC, ϕ_i, is active, it simultaneously functions as both: i) the full code of the basis state to which it was assigned, s_i; and ii) the entire explicit similarity distribution over ALL stored basis states, s_j, where the similarity of s_j and s_i is physically manifest as the fraction of ϕ_j's units that are active (as a subset of the single fully active code, ϕ_i). Assuming that for natural input spaces, similarity correlates with probability, we can restate the above as: the active code simultaneously functions as both: i) the single *most likely* basis state, ϕ_i; and ii) the explicit *probability distribution* over ALL stored basis states, where ϕ_j's probability is measured by the

(normalized) fraction of its units that is active. To restate for emphasis, a basis state is represented by a *set* of (physical) units being co-active and its probability is represented by the (normalized) fraction of those units active: in particular, complex numbers (PAs) are not needed.

If the sequence of input items are successive snapshots (e.g., frames of video) of a discrete dynamical system, then the transitions between the system's (basis) states, i.e., its dynamics, become embedded in the recurrent matrix during learning. Subsequently, in a retrieval/query mode, we can seed the field with a particular SDC and allow it to evolve, where the update from T to T+1 entails transmission of binary signals via the recurrent matrix from the SDC active at T, $\phi(T)$, resulting in a (generally new) SDC, $\phi(T+1)$. This transmission, which is the dominating step of the update operation: a) requires only a single iteration over the recurrent weights, the number of which, $(QK)^2$, is fixed; and b) does not depend on the number of SDCs (basis states) stored. That is, like the storage and best-match retrieval operations, updating the entire probability distribution over ALL basis states is also done in fixed time. In fact, each individual instance of the update operation, (from T to T+1) is identical to the best-match retrieval operation. It's just that, as explained above, since every SDC is both the code of a single state and an explicit probability distribution over ALL states, the same fixed time operation simultaneously retrieves (reactivates) the most likely state (i.e., most likely hypothesis) and updates the full probability distribution over all states. To my knowledge, this capability has not previously been described and I will argue that it meets the "exponential speedup" criterion of quantum computing, but in a purely classical setting. I will present simulation results demonstrating this capability.

To summarize, what we've done is change from representing basis states localistically, and their probabilities as *numbers* (the complex-valued PAs), which have *point* semantics and therefore do not support any physically realizable notion of superposition, to representing them in sparse distributed fashion, as *sets*, which have *extended-body* semantics (and do not reduce to numbers), and thus allow a straightforward physical, purely classical, realization of superposition. This change brings an understanding of reality that differs profoundly from the Copenhagen view in which, even though all possible basis states must somehow physically exist (otherwise they could not influence the evolution of the system), all but the single observed state are fundamentally inaccessible. In contrast, the model described here provides a simple classical explanation as to how any *partially active* basis state, i.e., at least one of its SDC's units is physically active, influences the next state of the system: it does so via the signals sent from its active unit(s) via the recurrent matrix.

Finally, I will also show that this model provides a simple classical realization / explanation of *entanglement*. Specifically, units become entangled when they are co-selected to be in an SDC, i.e., their entanglement is physically reified in the correlated changes made to their incoming and outgoing weights when they are activated in that SDC. For example, if two units, x and y [which, as stated earlier, may be arbitrarily distant in the field (in the physical memory)], were included in a previously stored SDC, ϕ_i, and if we know that x has been selected to be in the SDC, ϕ_t, currently being chosen at time t, then the correlated weight changes made to these units' weights when ϕ_i was active statistically influence (contain instantaneously available information about) the probability of y also being chosen for ϕ_t.

Rod Rinkus joined Purdue ECE's Center for Brain-Inspired Computing (C-BRIC) in May 2019 as a Lead Research Scholar. He is continuing his lifelong pursuit to understand the "neural code", i.e., how information is represented and processed in the brain—how memories are formed so quickly, last so long, and can be retrieved so reliably and rapidly (seemingly without serial search)—with the goal of working with other C-BRIC members to create novel, low-power hardware realizations leveraging the brain's principles and algorithms. After receiving his Ph.D. in Cognitive and Neural Systems from Boston University in 1996, he developed AI / Machine

Learning systems at several Boston-area research firms. He returned to academics in 2004 as Postdoc in computational neuroscience with Prof. John Lisman at Brandeis, revitalizing his thesis work on a sparse distributed representation (SDR) based neural model of memory and cognition. This eventually led to founding Neurithmic Systems in 2010, which was supported by ONR, DARPA, and commercial contracts, to further develop his SDR-based neuromorphic model, now called Sparsey, for application to spatiotemporal problems, e.g., recognition of events in video. Sparsey's very simple elements, i.e., binary units, weights, signals, and algorithms for learning and retrieval (inference), make it highly amenable to Non-von Neumann, processor-in-memory, hardware realizations, and closely aligned with C-BRIC's mission.