Random Projection in the Brain and Computation with Assemblies of Neurons

Christos H. Papadimitriou
Columbia University, USA
christos@columbia.edu

Santosh S. Vempala
Georgia Tech, USA
vempala@gatech.edu

Abstract

It has been recently shown via simulations [7] that random projection followed by a *cap* operation (setting to one the k largest elements of a vector and everything else to zero), a map believed to be an important part of the insect olfactory system, has strong locality sensitivity properties. We calculate the asymptotic law whereby the overlap in the input vectors is conserved, verifying mathematically this empirical finding. We then focus on the far more complex homologous operation in the mammalian brain, the creation through successive projections and caps of an assembly (roughly, a set of excitatory neurons representing a memory or concept) in the presence of recurrent synapses and plasticity. After providing a careful definition of assemblies, we prove that the operation of assembly projection converges with high probability, over the randomness of synaptic connectivity, even if plasticity is relatively small (previous proofs relied on high plasticity). We also show that assembly projection has itself some locality preservation properties. Finally, we propose a large repertoire of assembly operations, including *associate, merge, reciprocal project, and append,* each of them both biologically plausible and consistent with what we know from experiments, and show that this computational system is capable of simulating, again with high probability, arbitrary computation in a quite natural way. We hope that this novel way of looking at brain computation, open-ended and based on reasonably mainstream ideas in neuroscience, may prove an attractive entry point for computer scientists to work on understanding the brain.

2012 ACM Subject Classification Dummy classification

Keywords and phrases Dummy keyword

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.55

Funding This work was supported by NSF grants CCF-1563838, CCF-1819935, CCF-1763970, and CCF-1717349.

Acknowledgements Many thanks to Wolfgang Maass for many insightful discussions and exchanges during the early stages of our thinking in this area in general, and specifically about assembly operations, to Mike Collins for his insights regarding natural language in the human brain, and to Saket Navlakha for helpful comments on an early draft.

1 Introduction

The striking computational nature of the animal brain manifests itself even in the humblest circumstances. Flies sense odorants in their environment through specialized *olfactory receptor* neurons, of which there are roughly fifty different kinds. So, each smell is initially coded as a vector in 50 dimensions, where each coordinate is the level of activity of neurons

10th Innovations in Theoretical Computer Science Conference (ITCS 2019).
Editor: Avrim Blum; Article No. 55; pp. 55:1–55:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of each kind. Then a remarkable thing happens: This vector undergoes a *random projection* — a familiar ingredient of many algorithms, especially in connection to learning [6, 2, 22, 1, 3] — to a higher dimensional space. There is a 50×2000 sparse, and by all evidence [**?**] random, bipartite graph of synapses projecting the 50 kinds of olfactory receptors to a population of 2000 neurons called *Kenyon cells.* Next, the resulting 2000-dimensional vector of synaptic inputs undergoes an operation that is routine in neural systems: The activity of the Kenyon cells excites an inhibitory neuron, and the resulting activity of this neuron, at equilibrium, has the effect of increasing everybody's membrane potential, "turning off" all but roughly the 100 most active cells. We call this operation *cap*; it is also known as *k winners take all*, in this case with $k = 100$.

In a recent paper [7] it was shown empirically that this mapping, random projection followed by cap, has strong *locality sensitivity* properties (and therefore preserves similarity of smells, presumably to the animal's advantage), in fact outperforming in simulations certain variants of locality-sensitive hashing[1]. One of our results in this paper puts some mathematical teeth to this interesting empirical observation: We prove that if two binary vectors of the same sparsity overlap in a fraction α of their entries, and both undergo random projection to n dimensions followed by k-cap, then the two results will overlap in a fraction of about $(\frac{k}{n})^{\frac{1-\alpha}{1+\alpha}}$ (Theorem 1). For the small numbers of the insect brain ($\frac{n}{k} \approx \frac{2000}{100}$), this is substantial overlap that helps explain the empirical findings in [7] (see Figure 1).

In the mammalian brain numbers get roughly three orders of magnitude higher, and yet something similar seems to happen. Importantly, there is strong *recurrent synaptic connectivity* between excitatory neurons; that is, the random graph is now not just a directed bipartite graph, but the union of a bipartite directed graph and a non-bipartite directed graph interconnecting the receiving side (in contrast, synapses between the fly's Kenyon cells, if any, play no role there). In mammals, the random projection and cap operation does take place, but it is only the first step of a complex and sophisticated process, culminating in the creation of an *assembly of neurons.*

Assemblies. Already in 1949, neuroscience pioneer Donald Hebb predicted that memories and concepts are represented by tightly connected sets of neurons he called *assemblies*, whose near-simultaneous firing is tantamount to these concepts being thought about. During the last decade, it has been established experimentally [12, 13, 18], see also the survey [5], that such near-simultaneous firing of stable sets of neurons is an important part of the way the brain works. Assemblies have been hypothesized to underlie many of the higher cognitive operations in mammals, such as memory, reasoning, language, planning, etc., and yet, the way and manner in which this happens has not begun to be articulated; the computational framework of this paper is a first attempt at understanding how assemblies of neurons can carry out computation.

In our framework, the brain is divided into a bounded number of *brain areas.* Each brain area contains a number of excitatory neurons denoted by n; there are of course other neurons as well, for instance see the discussion on inhibition below. These excitatory neurons are interconnected in a sparse directed $G_{n,p}$ graph. Pairs of brain areas may also be connected, in one or both directions, through bipartite directed $G_{n,p}$ graphs[2].

Finally, the other two important aspects of our model are *cap* and *plasticity.* We assume

[1] As Alex Andoni notes (private communication, 2018), this is not true of the more advanced versions of LSH.

[2] See [16] for a technical discussion of *synaptic biases*, departures from the $G_{n,p}$ model noted in experiments, and the reasons why they may provide further support for the assembly hypothesis. We do not pursue this direction in the present paper.

that neurons fire — or do not — in discrete time steps (a very convenient and unrealistic assumption, which however does not interfere much with the rest of our framework). At each time and each brain area, the k out of n neurons that have largest synaptic input fire. That is, at time t for each neuron we add together the weights of the incoming synapses that originate in neurons (in the same or different area) which fired the previous time $t-1$, and select the k neurons out of the n in the brain area that have the largest sums. These are the neurons in the area that will fire at time t. The k-cap process is a simplification and approximation of the reality of *inhibition*, whereby an independent population of inhibitory neurons cause the excitatory neurons to have high enough membrane potential that an equilibrium at k firing neurons is quickly reached. Finally, plasticity: we assume that if there is a synapse from neuron i to neuron j, and neuron i fires at time t while neuron j at $t+1$, the weight of the synapse is increased by a factor of $1+\beta$ with $\beta > 0$; synaptic weights start at one, say[3]. Thus, the key parameters of our model are n, k, p, β, whose indicative intended values for the mammalian brain are, respectively, $10^7, 10^4, 10^{-3} - 10^{-2}, 10^{-1}$.

Defining Assemblies. An assembly is of course a set of neurons, in our framework all belonging to the same brain area. In past theoretical work [16] this is exactly how they were defined, a set of k neurons firing simultaneously. It is a highly interconnected set to ensure *stability,* that is, if enough neurons in it fire then soon all of them will[4] — and one of the main points of [16] was that there is a biologically plausible algorithm for selecting such a highly connected set of neurons in a sparse $G_{n,p}$ graph. These neurons might be poised to fire in a particular pattern, not necessarily all simultaneously as was assumed in [16] — and indeed, in our simulations, as well as in the literature on assembly simulations, one does see nontrivial patterns of firing. We believe the right way to define assemblies is as *distributions over the set of neurons in a Brain area whose support has size at most a fixed multiple of the cap size k.*

Projection. The most basic operation of assemblies is what we call *projection* — this is how assemblies are created and, once created, *copied* to other brain areas for further use. Assembly projection has been conjectured for a long time and has been established in several simulation papers [19, 17] and recently analytically proved [16] for a range of parameters. An assembly x in area A can project to a different area B, to which A has ample connectivity, creating a new assembly y; this operation is denoted $\texttt{project}(x, B, y)$. If in the future x is activated, y will follow suit; we say that $x = \texttt{parent}(y)$. We show that the operation $\texttt{project}(x, B, y)$ is carried out by assembly A simply *firing for a small number of steps*[5]. Once an assembly x has been created, its area is implicit, denoted by $\texttt{area}(x)$. To create an altogether new assembly y by $\texttt{project}(x, B, y)$, x must be a "proto-assembly," a set of neurons coding a world experience and residing at some higher area of the sensory cortex (such as the area IT of the visual cortex where whole objects are represented), projected to a non-sensory area admitting new assemblies (typically the hippocampus). One of our main results in this paper (Theorem 3) is that projection indeed works as described — with high probability, of course, with randomness supplied by the graph, and in fact for quite low

[3] There should also be a process of *homeostasis* which, at a slower time scale, keeps the sum of all weights from growing; but this aspect of the model, taken up in Section 5, does not affect the relative ordering of synaptic weights or sums thereof.

[4] This is one of the many important differences between this work and Valiant's pioneering theory of *items* from the 1990s [20, 21]

[5] $\texttt{project}(x, B, y)$ may seem superficially equivalent to an assignment $x = y$ in a programming language — except that, after such an assignment, variables x and y go on to live largely independent lives, whereas in assemblies x retains power over y, while y can only exist through x.

plasticity.

The projection process is quite intricate. It starts with the random projection plus k-cap described early in this introduction, creating a set of neurons that we call A_1, namely, the cells that happen to have the largest synaptic input from the projecting assembly x. We *assume* that the synaptic input of a neuron from assembly x is a Bernoulli random variable with parameters k, p and n samples. Notice also that, after the first round, the synapses between x and A_1 have been boosted by plasticity. As the projecting assembly keeps firing, cap will select the set of neurons A_2 that have highest *combined* synaptic input from x *and* A_1, and these will include two kinds of cells: the *core* neurons in $A_1 \cap A_2$, and new winners from outside A_1. What fraction of A_1 will become core? This is an important parameter of the situation, and we call it λ. To compute it, we set up an algebraic equation of Bernoulli expectations; as the expectation of a Bernoulli quantile depends explicitly on the fraction of winners, and concentration is strong, we can set up the equation and solve it in the "high probability" sense. For the parameter range of interest, λ is about half. Notice that, after this step, all synapses from x and A_1 to A_2 are boosted by plasticity.

Then the process is repeated, $A_3, A_4, \ldots, A_t, \ldots$, and we wish to show that $|B^*| = |\bigcup_t A_t|$ converges to some finite multiple of k (recall that this is our definition of an assembly). That is, eventually there will be a time after which there are no first-time winners. Unfortunately our already complicated Bernoulli analysis is no longer an option, for a variety of reasons. First, at time t the number of types of neurons grows exponentially with t: the type of each neuron is the set of τ's for which the neuron was in A_τ. In addition, the distribution of the synaptic input of neurons with complex type is not Bernoulli, because of conditioning. Instead, we resort to classifying each neuron by its *rough type at time t*, which is the number of *consecutive* times τ leading to $t-1$ during which the neuron was in A_τ. A crucial lemma states that the probability that the run will end at time t and the neuron will find itself outside A_t decreases exponentially with the length of the run (that is to say, the neuron's rough type), and in fact uniformly in t. Convergence to a union size that is a multiple of k (with a multiplier that is, naturally, a steeply increasing function of $\frac{1}{\beta}$) follows (Theorem 3). The proof is quite a bit easier in the *high plasticity regime* defined by $\beta > \sqrt{\frac{(1-p)\ln n}{pk}}$, in which case convergence is stronger in that the sequence A_t itself converges in finitely many steps (as indicated in [16]).

Operations on Assemblies. What is the right scale for understanding computation in the brain? We suspect that assemblies may underlie an important and powerful mode of brain computation, complementary to the computation involved in the processing of sensory input — heretofore the main focus of neuroscience. Such computation would encompass memory recall and association, deduction and reasoning, generating and parsing natural language, generating and manipulating stories and plans, even math. It happens at a level of abstraction intermediate between individual neurons and synapses at the lowest level, and whole brain computation at the highest; it is far more expressive than the latter, and much less cumbersome to describe than the former. In our quest to understand the full power of this mode of computation, in Section 5 we identify a repertoire of additional operations on assemblies, beyond projection. We only seek operations that are "realistic" in the following two orthogonal senses: (a) operations for which there is experimental evidence, in the sense that their existence would help explain extant experimental data, and which could possibly be themselves tested experimentally; and (b) operations which are in addition *plausible,* shown (analytically if at all possible, otherwise through simulations) to be realizable at the level of neurons and synapses in our framework. That is to say, each assembly operation must be "compiled down" to the level of neurons and synapses. Our list of operations includes, besides projection: *association,* in which two assemblies in the same area increase their intersection to reflect conceptual or statistical affinity — there is extensive experimental evidence for this operation, see [16] for an extensive discussion; *merge*, in which two assemblies from two different areas project to *the same new assembly* in a third area, an operation that seems important for processing syntax in natural language; *reciprocal project* (like project, except that the projected assembly is able to activate the original one, in addition to vice-versa); and *append,* an operation useful for creating and maintaining sequences. There are also several *control operations* allowing one to *read* the information of assembly activity in specific areas, or *disable* synaptic connectivity between areas — ultimately, to *write simple programs.* We show that this repertoire of assembly operations constitutes a programming system[6] which can simulate arbitrary computation in a way that is quite natural (Theorem 4). The point of this exercise is to demonstrate the power of this basis of primitives, not to hypothesize that the brain must function exactly this way.

Related work

Our work on assemblies is superficially related to (and was undoubtedly inspired by) Valiant's theory of *items.* There are stark contrasts between the two approaches: Assemblies are hypothesized to be densely connected, a requirement that makes their creation challenging, while items are ransom sets of neurons. And we believe that our model is far closer to the realities of the brain, as they are known now, than Valiant's; for one key difference, Valiant assumes plasticity (change in synaptic weights) to be arbitrarily programmable at the post-synaptic site, while we assume a very simple implementation of Hebb's rule. With this model we are able to address the problem of how the brain creates similar representations for similar stimuli.

Our earlier work on assemblies established experimentally the plausibility of projection and association [19], and theoretically so by relying on very high plasticity [16]. In this paper, we attack analytically the more realistic and considerably more challenging regime of small plasticity.

2 Model

We assume a finite number of brain areas, denoted by $A, B, \ldots$. Each brain area is a weighted directed graph whose vertices are n (think of n as 10^6 or 10^7) excitatory neurons, and whose edges are synapses between neurons; the positive weights vary dynamically through plasticity, see below. We assume that the edges are drawn from a $G_{n,p}$ distribution. That is, we assume that the probability of any edge is p and edges are chosen independently. In addition, between certain ordered pairs of areas (A, B) there is a $G_{n,p}$ directed bipartite graph from nodes of A to nodes of B. In other words, there is a finite directed graph with the areas as nodes, determining whether the two areas have synaptic connections. We assume that there is a mechanism to *disable* the synaptic connections between two areas A and B at any time.

We assume that events happen in discrete time steps (think of each step as about 20 ms). At each step t, every neuron i in every area A may or may not *fire.* Whether i fires depends on its *synaptic input* at time t. This is defined the sum over all neurons j that have synapses (j, i) (note that j can be either in area A or in an area B that does have synapses into A that are not disabled at time t). Denote this quantity as $SI(j)$. *We assume that neuron i in area*

[6] Which, to our credit, we refrained from dubbing "Assembly Language"...

A fires at time t if and only if $|\{j \in A : SI(j) \geq SI(i)\}| < k$, where k is a key parameter of the model (think of it as roughly $\sqrt{n}$). We call the set of neurons firing at a time t *the cap* of the area. The cap is a mathematically tractable way of capturing the important process of *inhibition*, whereby inhibitory neurons in an area (typically outnumbering excitatory ones) are excited by the firing of excitatory neurons in the area, and in response fire, preventing some excitatory neurons from further firing, and eventually reaching an equilibrium (called the *E-I balance* in the literature). Here we model this equilibrium by a constant k and ignore the transient.

The other important ingredient of our model is plasticity: We assume that if there is a synapse with weight w from neuron i to neuron j (either in the same area, or in another area with enabled synapses), and it so happens that i fires in time $t-1$ and j fires in time t, then the weight of synapse ij is in time $t+1$ equal to $w(1+\beta)$, where β (think of it as between 0 and 1, realistically at the lower end of this) is the plasticity coefficient. Plasticity is a very complex phenomenon with many important aspects and cases, but we feel that this simple rule (corresponding to Hebb's "fire together wire together" maxim) captures the essence of the matter reasonably well.

We shall elaborate certain further aspects of our model in the section on assembly operations.

3 The Overlap of Projections

In this and the next section we analyze how assemblies can be formed in our model. We assume that there is a *stimulus* A of k neurons firing in an area, with enabled synaptic projections to another area, where the assembly will be formed. We start with the simple case (modeling the insect brain) where A fires only once, forming the cap in the downstream area denoted $\text{cap}(A)$, and analyze how the overlap of two stimuli A and B is maintained in the process; note that here recurrent connections and plasticity do not get involved, and the weights can be thought to be one. The following observation will be useful: conditioning on a neuron not making it to a cap cannot increase its cap probability for future steps.

▶ **Lemma 1.** *Let A, B be two stimuli. Then for any node $i \in V$,*

$$\Pr(i \in \text{cap}(B) \,|\, i \notin \text{cap}(A)) \leq \Pr(i \in \text{cap}(B)) = \frac{k}{n}$$

where the probability is over the randomness of the graph.

Also, we will need the following well-known bound on the Gaussian tail.

▶ **Lemma 2** (Gaussian tail). *For $x \sim N(0,1)$ and $t > 0$,*

$$\frac{1}{\sqrt{2\pi}}\left(\frac{1}{t} - \frac{1}{t^3}\right)\exp(-t^2/2) \leq \Pr(x \geq t) \leq \frac{1}{\sqrt{2\pi}t}\exp(-t^2/2).$$

Now we state and prove our quantitative assessment of the locality sensitivity properties of the insect olfactory map pointed out empirically in [7].

▶ **Theorem 3.** *The expected overlap of the caps two stimuli that overlap in an α fraction of their nodes is*

$$\frac{|\text{cap}(A) \cap \text{cap}(B)|}{k} \gtrsim \frac{1}{(\ln(n/k))^{\frac{\alpha}{1+\alpha}}}\left(\frac{k}{n}\right)^{\frac{1-\alpha}{1+\alpha}}.$$

Proof. We bound the probability that any neuron i is in the cap of both A and B. For this, let x_i, y_i, z_i be the total input to node $i \in V$ from $A \setminus B, A \cap B$ and $B \setminus A$. Then $x_i, z_i \sim N((1-\alpha)kp, (1-\alpha)kp(1-p))$ and $y_i \sim N(\alpha kp, \alpha kp(1-p))$. Then, using the independence of $x_i + y_i$ and $z_i + y_i$ given y_i,

$$\begin{aligned}
&\Pr i \in \text{cap}(A) \cap \text{cap}(B)\\
&= \int\int\int \chi(x_i + y_i \in \text{top } k \text{ of } \{x_j + y_j\} \text{ and } z_i + y_i \in \text{top } k \text{ of } \{z_j + y_j\})\, d\gamma(x) d\gamma(z) d\gamma(y)\\
&= \int\int\int \chi(x_i + y_i \in \text{top } k \text{ of } \{x_j + y_j\} \,|\, y) \chi(z_i + y_i \in \text{top } k \text{ of } \{z_j + y_j\} \,|\, y)\, d\gamma(x) d\gamma(z) d\gamma(y)\\
&\geq \int \left(\int \chi(x_i + y_i \in \text{top } k \text{ of } \{x_j + y_j\} \,|\, y)\, d\gamma(x) \right)^2 d\gamma(y)\\
&\geq \int_{y_i} [\Pr(x_i \geq -y_i + kp + t \,|\, y_i)]^2\, d\gamma(y_i).
\end{aligned}$$

The last step above is the simple observation that a random draw $x_i + y_i$ from $N(kp, kp(1-p))$ is, with constant probability, in the top k of n iid draws from the same distribution if $x_i + y_i \geq \mathsf{E}(x_i + y + i) + t$ where $\Pr(x_i + y_i \geq t) \geq k/n$. The tail bound below shows that

$$t \sim \sqrt{(2\ln(n/k) - \ln(2\ln(n/k))kp}.$$

For convenience, we shift the distributions of x_i, y_i to $\bar{x} = (x - (1-\alpha)kp)/kp$ and $\bar{y} = (y - \alpha kp)/kp$ so that $\bar{x} \sim N(0, (1-\alpha))$ and $\bar{y} \sim N(0, \alpha)$. For $x \sim N(0,1)$, we will use the tail bound in Lemma 2:

$$\frac{1}{\sqrt{2\pi}} \left(\frac{1}{t} - \frac{1}{t^3} \right) \exp(-t^2/2) \leq \Pr(x \geq t) \leq \frac{1}{\sqrt{2\pi} t} \exp(-t^2/2).$$

Thus, for any $\alpha < 1$,

$$\begin{aligned}
&\Pr(i \in \text{cap}(A) \cap \text{cap}(B))\\
&\geq \int_{\bar{y}} \Pr_{\bar{x}} (\bar{x} \geq -\bar{y} + t)^2 \, d\gamma(\bar{y})\\
&\geq \int_{\bar{y}} \frac{1}{2\pi(1-\alpha)} \min\left\{ \frac{1-\alpha}{(t-\bar{y})^2}, 1-\alpha \right\} \exp\left(-2\frac{(t-\bar{y})^2}{2(1-\alpha)} \right) \frac{1}{\sqrt{2\pi\alpha}} \exp\left(-\frac{\bar{y}^2}{2\alpha} \right) d\bar{y}\\
&\geq \left(\frac{1}{2\pi t^{2/(1+\alpha)}} \exp\left(-\frac{t^2}{1+\alpha} \right) \right) \int_{\bar{y}} \frac{t^{2/(1+\alpha)}}{\sqrt{2\pi\alpha}} \min\left\{ \frac{1}{(t-\bar{y})^2}, 1 \right\} \exp\left(-\frac{(\bar{y} - \frac{2\alpha}{(1+\alpha)} t)^2}{2\alpha(1-\alpha)/(1+\alpha)} \right) d\bar{y}\\
&\geq \sqrt{\frac{1-\alpha}{1+\alpha}} \left(\frac{k}{n} \right)^{\frac{2}{1+\alpha}} \frac{1}{t^{2\alpha/(1+\alpha)}} \int_y \frac{\min\left\{ \frac{1}{\left(\frac{1-\alpha}{1+\alpha} - \frac{y}{t} \right)^2}, 1 \right\}}{\sqrt{2\pi\alpha(1-\alpha)/(1+\alpha)}} \exp\left(-\frac{y^2}{2\alpha(1-\alpha)/(1+\alpha)} \right) dy\\
&\geq \sqrt{\frac{1-\alpha}{1+\alpha}} \left(\frac{k}{n} \right)^{\frac{2}{1+\alpha}} \frac{1}{t^{2\alpha/(1+\alpha)}} \int_y \frac{1}{\sqrt{2\pi}} \min\left\{ \frac{1}{\left(\frac{1-\alpha}{1+\alpha} - \frac{y}{t} \sqrt{\frac{\alpha(1-\alpha)}{1+\alpha}} \right)^2}, 1 \right\} \exp\left(-\frac{y^2}{2} \right) dy\\
&\geq \frac{\sqrt{\frac{1-\alpha}{1+\alpha}}}{(2\ln(n/k))^{\alpha/(1+\alpha)}} \left(\frac{k}{n} \right)^{\frac{2}{1+\alpha}}.
\end{aligned}$$

Thus the expected fraction of overlap is this probability times n divided by k, i.e.,

$$\Omega\left(\frac{1}{(\ln(n/k))^{\frac{\alpha}{1+\alpha}}}\left(\frac{k}{n}\right)^{\frac{2}{1+\alpha}}\frac{n}{k}\right)=\Omega\left(\frac{1}{(\ln(n/k))^{\frac{\alpha}{1+\alpha}}}\left(\frac{k}{n}\right)^{\frac{1-\alpha}{1+\alpha}}\right).$$

◀

It seems that the steps in this proof, including the suppression of constants in the end, are quite parsimonious, in that the stated lower bound is not very far from the truth. In Figure 1 we compare our bound with simulations of the map for various values of α and with $n/k = 2000/100 = 20$ (the values that pertain to insect olfaction) and $n = 10^4, k = 100$, and also to our bound without the logarithmic factor.

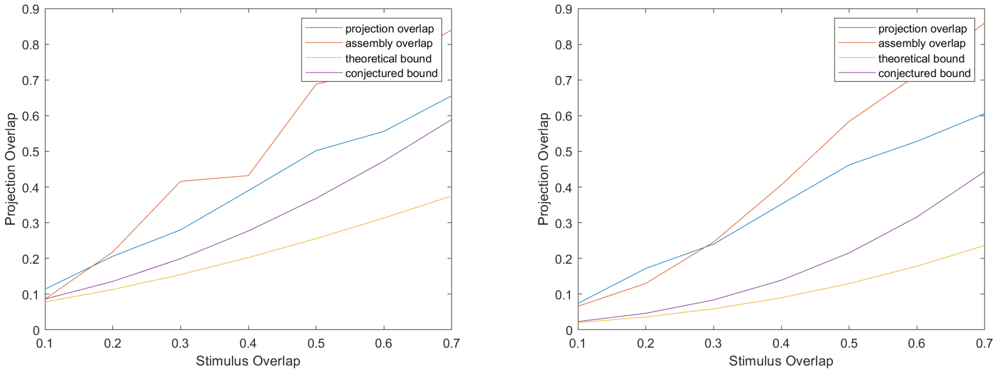

Figure 1 The first figure is with $n = 2000, k = 100$ and the second with $n = 10000, k = 100$; each empirical plot is the average of 5 independent trials. For the assembly creation we used plasticity of $\beta = 0.1$. The theoretical bound plotted is $(k/n)^{(1-\alpha)/(1+\alpha)}/\ln(n/k)^{\alpha/(1+\alpha)}$, while the conjectured bound is the same without the log factor.

4 Bounding the Support of an Assembly

In this section we turn to assemblies in the mammalian brain, in which recurrent synapses and plasticity become important. We assume that a stimulus consisting of $k \geq \sqrt{n}$ neurons in an upstream area fires repeatedly. The cap at $t = 1$, denoted A_1, which was analyzed in the previous section, is only the preamble of a complex process. At $t = 2$ the stimulus fires again, and now the area receives combined input from the stimulus *and* from A_1. A cap denoted A_2 will be formed, probably containing a considerable part of A_1 but also *first-timers* (by which we mean, neurons not heretofore participating in any cap). Meanwhile, plasticity has changed the weights. The process is repeated a number of times, with new winners displacing some past winners from the new cap, while plasticity acts in a stabilizing way. Convergence — that is, $A_t = A$ for all $t > t_0$ — cannot be guaranteed with high probability (experiments show some periodic-like movement of neurons, without any new first-timers). The interesting question is, will the process converge, in that after some point and after there will be no new winners? (Recall that this is what we mean by an assembly, a set of neurons of size a small multiple of k firing in a pattern.). If so, we are interested in the size of the assembly's *support,* the union of all the A_ts. The bound on the support depends crucially on the plasticity parameter β, with high plasticity leading to small support (close to the cap size k) but even very small positive plasticity leading to bounded support size (a fact that is harder to prove). We denote by A^* the union of $A_0, A_1, A_2, \ldots$.

▶ **Theorem 4** (High Plasticity). *Assume that the plasticity parameter $\beta \geq \beta_0 = \frac{(\sqrt{2}-1)\sqrt{\ln n}+\sqrt{2}}{\sqrt{pk}+\sqrt{\ln n}}$. Then WHP the total support of the assembly can be bounded as*

$$|A^*| \leq k\frac{1}{1-\exp(-(\frac{\beta}{\beta_0})^2)} \leq k + O\left(\frac{\ln n}{p\beta^2}\right).$$

Proof. Let $\mu_1 = 1, \mu_2, \ldots, \mu_t, \ldots$ be the fraction of first-timers in the cap at step t. The process stabilizes when $\mu_t < 1/k$. Using the tail bound of the Gaussian, since the new winners must be in the top $\mu_t k$ of remaining $n - k \sim n$ neurons, the activation threshold at step t is therefore very close to

$$C_1 = pk + \sqrt{2pk \ln \frac{n}{k}}, \quad C_t = 2pk + 2\sqrt{pk \ln \frac{n}{\mu_t k}} \text{ for } t \geq 2.$$

Note that the mean term is pk for the first step and $2pk$ for all subsequent steps since the number of neurons firing is the k stimulus ones plus k from the brain area.

First consider a neuron that make it to the first cap. To bound the probability that that it will remain in the next cap, we note that at this point, the total activation from the input synapses is at least $(1+\beta)C_1$ and from the recurrent synapses it is at least X where $X \sim N(pk, p(1-p)k)$ is the signal from the recurrent synapses coming from nodes in the first cap. In order for a node to remain in the next cap, we need that

$$(1+\beta)C_1 + pk + X \geq C_2$$

where now $X \sim N(0, p(1-p)k)$. Substituting for C_1, C_2, and using $L = 2\ln(n/k)$, and μ as the fraction of first-timers in the second cap, we have

$$\begin{aligned}\Pr(j \in C_2 \,|\, j \in C_1) = 1 - \mu \quad &\geq \quad \Pr(X \geq -\beta pk - (1+\beta)\sqrt{pkL} + \sqrt{2pk(L + 2\ln(1/\mu))})\\ &\geq \quad \Pr(X \geq -\beta\sqrt{pk} + \sqrt{2(L + \ln(1/\mu))} - (1+\beta)\sqrt{L})\\ &\quad\quad \text{rescaling so that } X \sim N(0,1).\\ &\gtrsim \quad 1 - \exp\left\{-(\beta\sqrt{pk} + (1+\beta)\sqrt{L} - \sqrt{2(L+\ln(1/\mu))})^2/2\right\}.\end{aligned}$$

In other words,

$$\sqrt{2\ln(1/\mu)} \leq \beta\sqrt{pk} + (1+\beta)\sqrt{L} - \sqrt{2(L + \ln(1/\mu))}.$$

Now setting

$$\beta \geq \beta_0 = \frac{(\sqrt{2}-1)\sqrt{L}+\sqrt{2}}{\sqrt{pk}+\sqrt{L}}$$

gives $\mu < 1/e$, i.e., the overap with the next cap is at least a $1 - (1/e)$ fraction. The probability of remaining in the cap rapidly increases with the number of consecutive times a neuron stays in the cap. To see this, suppose neuron j enters the cap for the first tiema at time t, by exceeding the threshold C_t and stays for i consecutive caps (including C_t. The, to stay in the next cap, it suffices that

$$(1+\beta)^i C_1 + pk + X \geq C_{i+1}$$

where $X \sim (0, p(1-p)k)$. Then, rescaling so $X \sim N(0,1)$,

$$\begin{aligned}\Pr(j \in C_{i+1} \,|\, j \in C_1) \quad &= \quad 1 - \mu\\ &\geq \quad \Pr(X \geq (1 - (1+\beta)^i\sqrt{pk} - (1+\beta)^i\sqrt{L} + \sqrt{2(L + 2\ln(1/\mu))})\\ &\gtrsim \quad 1 - \exp\left\{-(i\beta\sqrt{pk} + (1+i\beta)\sqrt{L} - \sqrt{2(L+\ln(1/\mu))})^2/2\right\}.\end{aligned}$$

Rewriting,

$$\sqrt{2\ln(1/\mu)} + \sqrt{2(L+\ln(1/\mu))} - \sqrt{L} \le i\beta(\sqrt{pk}+\sqrt{L})$$

or

$$\beta \ge \frac{1}{i} \cdot \frac{\sqrt{2\ln(1/\mu)} + \sqrt{2(L+\ln(1/\mu))} - \sqrt{L}}{(\sqrt{pk}+\sqrt{L})}$$

which is less than β_0 for $\mu = e^{-i^2}$.

Next we consider a new first time winner in round t. In order for this neuron to make it to the cap at time $t+1$, we need that

$$(1+\beta)\frac{(2-\mu)}{2}C_t + \mu pk + X \ge C_{t+1}$$

where $\mu = \mu_{t+1}$ is the fraction of newcomers in the next cap and $X \sim N(0, \mu p(1-p)k)$. Rescaling so that $X \sim N(0,\mu)$, we have $\Pr(j \in C_{t+1} \mid j \in C_t)$ is

$$1-\mu \ge \Pr(X \ge -\beta(1-\frac{\mu}{2})2\sqrt{pk} - (1+\beta)(1-\frac{\mu}{2})\sqrt{2(L+\ln(1/\mu_t))} + \sqrt{2(L+\ln(1/\mu))})$$

Using the tail bound and rewriting as before, we have

$$\beta \ge \frac{2\ln(1/\mu) + \frac{\mu}{2}\sqrt{2(L+\ln(1/\mu_t))} + \frac{\ln(\mu_t/\mu)}{L}}{(1-\frac{\mu}{2})(2\sqrt{pk}+\sqrt{2(L+\ln(1/\mu_t))})}$$

which is less than β_0 for $\mu = \mu_t/e$. In other words, the β threshold to do this and ensure that μ drops by a constant factor is lower than the threshold β_0 for the first step. Finally, as before, the probability of staying in the cap increases rapidly with the length of the neurons' winning streak.

If $\beta \ge \beta_0$, then μ_t drops off exponentially. i.e., the probability of leaving the cap once in the cap for i consecutive times $1 - p_i^t$ drops off exponentially. Using these facts, we get

▶ Claim 1.

$$\prod_{i\ge1} p_i \ge \prod_{i\ge1}(1-\exp(-i^2(\frac{\beta}{\beta_0})^2)) \ge \frac{1}{2}.$$

The claim gives a lower bound on the probability that a neuron that makes it to a cap for the first time remains in the cap for all future times. As a result, each neuron that makes it a cap for the first time has a probability of at least $q = 1-\exp(-(\frac{\beta}{\beta_0})^2)$ of remaining in all future caps. Thus, the total support of all caps together is at most k/q in expectation. This completes the proof of the theorem. ◀

We now turn to the regime of low plasticity, including zero plasticity. The bounds here will be higher asymptotically, as reflected also in our experiments (see Figure 2). We note however that for parameter ranges of interest for the brain, e.g., $n = 10^6, k = 10^3$,

$$\left(\frac{n}{k}\right)^{1/4} < \ln(n/k).$$

The guarantees below are meaningful and nontrivial only when k is sufficiently large as a function of n.

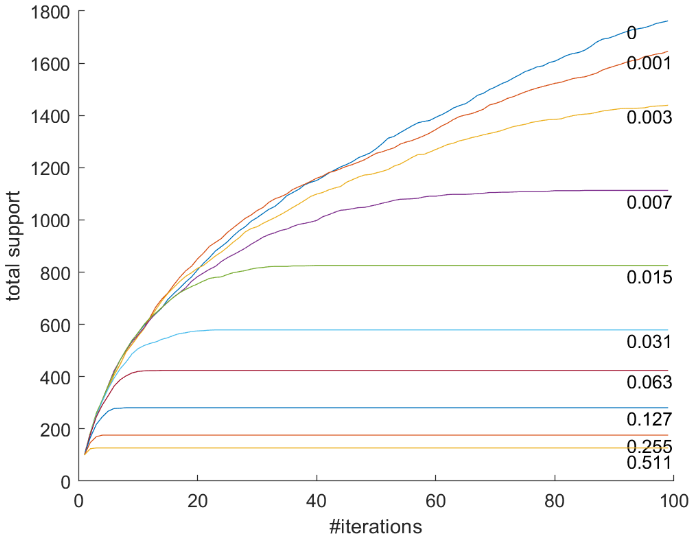

Figure 2 The total support size at different values of plasticity β ranging from 0 to just over 0.5 for a random network with $n = 10^4$ neurons, edge probability $p = 0.01$ and assembly size $k = 100$. The x axis is the number of iterations.

▶ **Theorem 5** (Low Plasticity). *Let a network with n nodes have edge density p, plasticity parameter β, and cap size $k \geq \sqrt{n}$. For a sequence of caps $A_0, A_1, A_2, \ldots\ldots A_t, \ldots$, let A^* be their union. Denote $\mu = \sqrt{k/n}$. Then,*

1. *for $\beta = 0$,*

$$\mathsf{E}\left(|A^*|\right) \leq k\left(\frac{1}{\mu}\right)^{\frac{1}{\mu}}.$$

2. *for $\beta > 0$,*

$$\mathsf{E}\left(|A^*|\right) \leq k\left(\frac{1}{\mu}\right)^{\frac{1}{2\beta}}.$$

Proof. For the first part, let $\mu_0, \mu_1, \ldots, \mu_t, \ldots$ be defined as $\mu_0 = 0$ and

$$\mu_t = \frac{|A_t \cap A_{t-1}|}{k},$$

the fraction of the cap that persists to the next step.

We will show that the expected values of μ_t form an increasing sequence and give a recursive lower bound. To get a lower bound on μ_1, for a neuron j, let x be the total signal from the stimulus and y from A_0, normalized, i.e., $x, y \sim N(0,1)$. Then,

$$\begin{aligned}
&\Pr(j \in A_1 \,|\, j \in A_0) \\
&\geq \Pr(x + y \geq 2\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))} \,|\, x \geq \sqrt{2\ln(n/k) - \ln(2\ln(n/k))}) \\
&\geq \Pr(y \geq (2-\sqrt{2})\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \mu_0 = \left(\frac{k}{n}\right)^{-(\sqrt{2}-1)^2}.
\end{aligned}$$

For general $t > 1$, let x be the signal from the stimulus y from the overlap $A_t \cap A_{t-1}$ and z from the rest of A_t. Then, with $z \sim N(0, (1-\mu_t))$,

$$\begin{aligned}
\mu_{t+1} &= \Pr(j \in A_{t+1} \,|\, j \in A_t) \\
&\geq \Pr(x + y + z \geq 2\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))} \,|\, x \geq \sqrt{2\ln(n/k) - \ln(2\ln(n/k))}, \\
&\quad \text{and } y \geq \mu_t(2-\sqrt{2})\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \Pr(x \geq (2-\sqrt{2})(1-\mu_t)\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \left(\frac{k}{n}\right)^{-(\sqrt{2}-1)^2(1-\mu_t)} \\
&= \mu_0^{1-\mu_t}.
\end{aligned}$$

The probability that a neuron j, which enters the cap at the first step, stays in the cap is thus at least

$$\begin{aligned}
\prod_t \mu_t &\geq \mu_0 \cdot \mu_0^{1-\mu_0} \cdot \mu_0^{1-\mu_0^{1-\mu_0}} \cdot \ldots \\
&= \mu_0^{1+(1-\mu_0)+(1-\mu_0^{1-\mu_0})+\ldots} \\
&\geq \mu_0^{1+(1-\mu_0)+(1-\mu_0)^2+(1-\mu_0)^3+\ldots} \\
&= \mu_0^{\frac{1}{\mu_0}}
\end{aligned}$$

where we used the fact that $1 - \mu_0^{(1-\mu_0)^i} = 1 - (1-(1-\mu_0))^{(1-\mu_0)^i} \geq (1-\mu_0)^{i+1}$.

So far, the computation was only for neurons that were in the very first caps. For neurons that make their first entrance later, the calculation is a bit different. Suppose a neuron enters the cap for the first time at iteration t. For general $t > 1$, let x be the signal from the stimulus y from the overlap $A_t \cap A_{t-1}$ and z from the rest of A_t. Then, with $z \sim N(0, (1-\mu_t))$, noting that x, y make up $(1+\mu_t)/2$ of the threshold C_t,

$$\begin{aligned}
&\mu_{t+1} \\
&= \Pr(j \in A_{t+1} \,|\, j \in A_t) \\
&\geq \Pr(x + y + z \geq 2\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))} \,|\, x + y \geq (1+\mu_t)\sqrt{\ln(n/k) - \ln(2\ln(n/k))}) \\
&\geq \Pr(x \geq (1-\mu_t)\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \left(\frac{k}{n}\right)^{-(1-\mu_t)/2} \\
&= \mu^{1-\mu_t}.
\end{aligned}$$

Note that μ here is smaller than μ_0 for neurons that enter in the first cap. The computation for later steps, for such a neuron is similar, and we get that the probability that such a neuron stays in the cap forever is

$$\prod_t \mu_t \geq \mu \cdot \mu^{1-\mu} \cdot \mu^{1-\mu^{1-\mu}} \cdot \ldots \geq \mu^{\frac{1}{\mu}}$$

as before. This completes the first part for $\beta = 0$.

For the second part, with $\beta > 0$, the calculation follows the same outline, except that the signal from the input is boosted by a factor of $(1+\beta)$ in each iteration, and the signal from previous caps is boosted by $(1+\beta)$ for a diminishing fraction $\prod_t \mu_t$. Ignoring the latter boost (for a lower bound),

$$\begin{aligned}
\mu_{t+1} &\geq \Pr(x+y+z \geq 2\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))} \,|\, x \geq \sqrt{2\ln(n/k) - \ln(2\ln(n/k))}, \\
&\quad \text{and } y \geq \mu_t(2-\sqrt{2})\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \Pr(x \geq (2-\sqrt{2}(1+\beta)^t)(1-\mu_t)\sqrt{\ln(n/k) - 0.5\ln(2\ln(n/k))}) \\
&\geq \left(\frac{k}{n}\right)^{-(\sqrt{2}-(1+\beta)^t)^2(1-\mu_t)} \\
&= \mu^{(1-t\beta)(1-\mu_t)}.
\end{aligned}$$

We can now lower bound the probability of a neuron staying in the cap once it enters, and thereby the expected size of the total support. ◀

Locality Sensitivity of Assemblies. Returning to the motivating story on fly olfaction, is the assembly projection operation as locality sensitive as the simpler variant in insects? It appears that overlap of assemblies is an important indication of affinity of various sorts (co-occurrence, correlation, connection, similarity, etc.), and thus it matters whether or not it is preserved in projection. What we are able to show is that, if two sets of k cells overlap in a fraction of α, and these two sets are projected sequentially to the same brain area, *the cores* of two resulting assemblies will share at least λ^2 fraction of the overlap of their initial projections (given by Theorem 3); recall that λ is the size of the core over k, and for the parameters of interest is about half. Such a modest overlap at the core — the best connected part of the assembly — is a good omen for a large overlap of the two assemblies that will eventually emerge, an intuition that is supported by simulations, see Figure 1.[7]

5 Computing with Assemblies

The assembly hypothesis proposes that assemblies are the standard representations used in higher brain functions — memory, language, reasoning, decision-making, planning, math, music, story-telling and discourse — suggesting a grand and mysterious computational system with assemblies at its center, its basic data type. *How does this computational system work?* Foremost, what are its elementary operations?

- Assemblies do appear to *project* (see the discussion in [11] for an inspiring description of the process in the mouse piriform cortex): this is about the only way that assemblies can be created, and projection appears to be a most useful operation — in fact, in its absence, it is hard to imagine what assemblies may be good for. We denote the operation of an assembly x projecting to area A to create a new assembly y as `project`(x, A, y) (the area of assembly x, denoted `area`$(x) \neq A$, is implicit). Henceforth, `parent`$(y) = x$[8]. Through `project`, arbitrary relations can be maintained, with brain areas being the columns and time steps the rows; for example, a recent experiment [10] seems to suggest that the "subject-verb-object" relation in natural language may be achieved this way.

[7] We can prove something weaker, namely that substantial overlap persists to the assemblies, albeit only for sufficiently high plasticity, and under the additional assumption that the synaptic weights from the first projection have "faded" enough by homeostasis.

[8] As we shall see, some operations such as `reciprocal-project` make the `parent` function ambiguous, but we shall be ignoring this issue here.

- We also know from experiments [14, 8] that assemblies *associate* by exchanging cells (apparently a few percentage points of their support) when they become related through co-occurrence in the world and perhaps through other acquired relations. We denote this by `associate`(x, y) — x and y should of course be in the same area. It can be provably carried out by activating `parent`(x) and `parent`(y), assumed to be in different areas, for a few steps [16]. It is natural to hypothesize that cell sharing between x and y has the effect that y may be henceforth activated, with some non-zero probability, when x is activated, and vice-versa. This opens up intriguing possibilities of sophisticated probabilistic reasoning and programming, and we suspect that much of the power of the assembly model may lie in this direction — which however we do not explore or exploit here.
- On another front, recent fascinating experiments [9, 23, 24, 15] suggest that *language processing* in humans involves the building and maintenance of syntactic structures such as syntax trees, and it is natural to assume that assemblies representing words are implicated there as well. We postulate the operation `merge`(x, y, A, z) which takes two assemblies x, y in different areas, and projects them *both* to assembly z in a third area A. Merge, the ability to consider two things as one, has been hypothesized in linguistics to be the quintessence of syntax, see for example [4]. It follows from the results in this paper that it can be implemented in our framework.
- A more complex and very useful operation is `reciprocal-project`(x, A, y, B, z) which creates in two areas A and B two assemblies y and z that can activate one another (while y can activated by x, as in ordinary `project`). It is assumed that there is synaptic connectivity from `area`(x) to A and both ways between A and B. The original assembly x, residing in a third area, can activate directly y. We conjecture that this operation can be carried out in our framework with high probability; it works reliably in simulations. `reciprocal-merge` is a straightforward generalization, which seems useful for language generation. Finally, another related operation is `append`(x, A, y), useful for creating sequences, which we do not detail here.

5.1 The Power of Computation with Assemblies

According to the assembly hypothesis, assemblies and their operations are crucial for higher mental activities such as planning, language, and reason. The question may then arise: Is this purported computational system powerful enough? In particular, *is it Turing complete?* Many computer scientists are by instinct dubious about the value of such a pursuit; we agree, and in addition we are convinced that, if the assembly hypothesis is correct, the computational power of assemblies is wielded through means that are orthogonal to computer programming. On the other hand, an assessment of the computational power of this system can usefully inform our modeling, and in particular our search for essential primitives.

To continue on this path, we must create a programming system, formal enough to address the Turing completeness question, for writing simple programs with lines such as

`if area`$(y) = A$`, project(parent`$(y), B, z$`).`

To this end, we need to assume an environment in which names of assemblies, once declared — typically in a command such as `project`(x, A, y) — can be used in subsequent steps of the same program (area names are finite and fixed). Also, we introduce certain new primitives: `activate`(x) simply activates assembly x for a few steps; that is, we assume that `project` creates as a side-effect a *fuse* that can activate the new assembly. Also, we assume that the downstream synapses from area A to area B are by default inactive, and must be activated explicitly by the operation `enable`(A, B). To illustrate, `project`(x, A, y) is almost equivalent to

`enable(area`$(x), A$`); repeat` T `times: activate(`x`); disable(area`$(x), A$`),`

missing only a mechanism that names the new assembly y. Here T is the number of spikes required for assembly projection (about a dozen in simulations). Of course, it is debatable how realistically one expect such a programming framework to be operating in the brain.

We also introduce a *read* operation[9] returning information about the assemblies that are presently active, and their areas. Notice that all this assumes a simple computational mechanism acting as an *interpreter,* and lying outside our framework[10].

Finally, we must address the issue of *reliability* in assembly computation. We shall make some assumptions:

- Any newly created assembly is a *random* set of $k = \gamma\sqrt{n}$ neurons in its area.
- Two assemblies can interfere destructively in their operations, for example by spurious associations between them, but only if they overlap in more than $\epsilon\sqrt{n}$ cells; the literature seems to suggest that ϵ is at least 1%.
- At last we need to introduce *homeostasis:*. We assume that synaptic weights *fade* with time, regressing to the value 1. That is, at every time step weight w becomes $\max\{\frac{w}{(1+\beta')}, 1\}$, where $0 < \beta' << \beta$, the plasticity parameter.[11]
  Fading is both realistic and necessary for the simulation, since in its absence the computational system cannot erase information, and is therefore severely limited.
- Fading means that eventually all assemblies will lose their synaptic density and connection with their parent. To prevent this, we introduce *permanent versions* of operations such as `project`. For example, `permanent_project(`x, A, y`)` involves, besides executing n ordinary `project` operation, repeating `activate(`x`)` every τ steps (with synaptic connections between the two areas in focus enables), where τ is a small constant, much smaller than $\frac{\beta}{\beta'}$, either indefinitely or until an explicit `fade(`y`)` command. There is evidence that such processes do happen in the brain, for example by fading, or reviving through rehearsal raw memory traces in the hippocampus.

The following is needed in the proof of the main result:

▶ **Lemma 6.** *The probability that a new assembly will interact destructively with a particular already existing assembly in the same area is at most* $\exp(-\frac{\epsilon\sqrt{n}}{\gamma^2})$.

▶ **Theorem 7.** *The computational system described above can correctly simulate arbitrary* $O(\sqrt{n})$*-space computations with probability* $1 - \exp(O(\sqrt{n}))$.

Sketch: A Turing machine with a one-way circular tape of length $m = O(\sqrt{n})$, tape alphabet Σ and state set K can be simulated by a program of assembly operations. Let us assume the input-output convention that a new assembly appears in one of two designated input areas I_0, I_1 at designated and well separated times, encoding a binary input tape; and that, upon accepting termination, an assembly will appear in another area T. The Turing machine will be simulated by $|\Sigma| + |K| + 6$ brain areas: the three input-output areas I_1, I_0, O, two areas for representing the tape denoted T_1 and T_2, one area for representing the current state,

[9] Following a suggestion by Buszáki [5] that assemblies must be accompanied by a reader mechanism — as Buszáki puts it: “if a tree falls in the forest and there is nobody around to hear it fall, has it really fallen?”

[10] We do realize this is a strong assumption, unlikely to be literally true; we expect that the computational power of assemblies is realized through more organic means

[11] An equivalent, and perhaps more realistic, model of homeostasis would be to normalize the incoming weights of each neuron separately.

denoted S, plus one area for each tape symbol a and state q, denoted, respectively, L_a and S_q. See Figure 3.

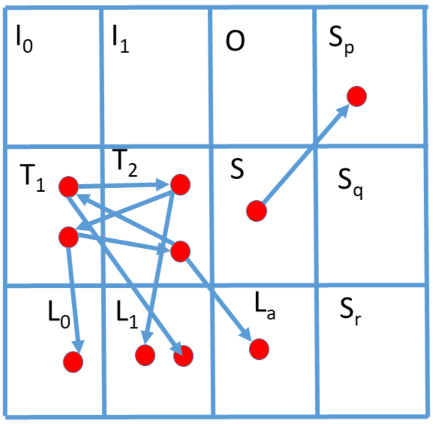

Figure 3 Our representation of configuration (state, circular tape contents) $[p, 011a]$

In the input phase, while the input is read from either I_0 or I_1 (depending on whether the input symbol is 0 or 1, assumed both to be in Σ (recall the input-output conventions), a chain of assemblies is created projecting back and forth between the two T_i areas (see Figure) through *permanent* `project` operations.

Each assembly in these two areas represents a tape square. The current symbol a in this square is represented through a projection to an assembly in area L_a, a projection that is permanent until it is explicitly faded when the same tape square is scanned again.

Similarly, another standard assembly s in area S points, through a projection (non-permanent, since the state changes at every step), to an area S_q representing the current state q (initially the starting state). The synapses from S to S_q are enabled, while the synapses from S to all other S_p's are not[12].

When the square corresponding to an assembly x, in one of the areas T_1, T_2, is scanned by the tape head, then x and s fire and a `read` is issued. Depending on the areas where assembly activity is read, say S_q and L_a, the correct current symbol a and state q are identified. Suppose that Turing machine's transition is $\delta(q, a) = (p, b)$. The synapses from S to S_q are disabled and those to S_p enabled, the assembly representing the previous symbol q is faded, and `permanent_project`(x, L_b, y) is executed to record the current symbol of the tape square represented by x; similarly for state. Then x fires again and a read is issued, to identify the tape assembly corresponding to the tape square that is next, and the computation continues. The straightforward details are omitted.

[12] Notice that this effectively stores the state in the current instruction of the program; it can be done in more natural ways.

6 Discussion and open questions

We have identified a basic computational operation — random synaptic projection to a brain area followed by the selection, through inhibition, of the k neurons with the highest synaptic input — that appears to be ubiquitous in the animal brain and also useful for implementing more complex operations, but also happens to be mathematically concrete, productive, and interesting. Assembly projection can be the basis of a computational system at an intermediate level of abstraction — and unlike anything else that we have seen in theoretical neuroscience. Such a system, we hypothesize, may underlie the higher mental functions of the human brain — not an intensely researched subject in neuroscience. This hypothesis must be pursued both analytically, and — importantly — experimentally. We also believe that this line of work, and the rather simple and concrete model of brain operation it entails involving distinct brain areas, random graph connections, inhibition through cap, and probabilistic analysis, may constitute a promising entry point for theoretical computer scientists who want to work on brain-related problems. One of the contributions of this paper is pointing out the locality sensitive nature of assembly projection; this, together with the computational nature of association (which we did not consider here) promise to be important future directions for this work.

Assemblies may be implicated in implementing *natural language* in the human brain. Many recent experimental papers, see [23, 24, 10, 15, 9] among many others, appear to suggest that assembly-like operations like `projection` and `merge` may be implicated in language generation and processing.

We conclude with some more precise questions, that are motivated directly by our findings, and will help solidify the mathematical theory of assemblies, some of which we have already discussed in context in this paper.

1. Assembly support size. Is there a phase transition in the support size of an assembly (from $\omega(k)$ to $k + o(k)$) as the plasticity parameter β increases?
2. Assembly convergence. For high plasticity and with high probability, the limit of the random project plus cap process is a single fixed subset of size k. What are other possible limiting behaviors? E.g., is it possible to get two subsets of size k (possibly overlapping) that fire alternately? (We know cases where this happens at a small scale, that is, the two subsets of size k differ in 1-3 cells.) Will the limit have a common core (of what size as a function of plasticity) that always fires? Is the limit an activity pattern of finite length/description?
3. Model. Can our results be extended to less stylized models in which neurons fire asynchronously, or there is explicit inhibition (instead of cap)?
4. Base graph. We have assumed the base graph to have independently chosen edges. What is a deterministic condition on the base graph that suffices? E.g., is it enough to have expansion and roughly uniform degrees? Is global expansion necessary or do sufficiently strong local properties suffice (e.g., degree and co-degree)?
5. Extending GNP. Are richer models, e.g., those with higher reciprocity or triangle density, useful? For example, do they enable more powerful or efficient computations?
6. Computational power. Show that randomized $s(n)$ space bounded computation can be simulated with n neurons and $O(1)$ brain areas for some function $s(n)$ larger than $\sqrt{n}$.
7. Capacity. Suppose that, in a brain area, we want to maintain with high probability pairwise intersections: two assemblies that intersect in a large (α or more, say) fraction of their support should continue to so intersect, and similarly for pairs that intersect in less

than α fraction. For how many assemblies can we guarantee this invariant, as a function of n?

8. Learning. Can assemblies perform learning (supervised or unsupervised)? Simulations suggest that assemblies can learn well-separated half-spaces quite naturally. Can this be proved formally? And what more ambitious forms of learning through assemblies are possible?
9. Assemblies vs 1-step Projections. Are assemblies (created as the limit of iterated random-project-and-cap) better for learning than 1-step (insect-like) projections? Is the recurrence of the mammalian brain a bonus or a handicap for learning?
10. Articulate a brain architecture for syntax (the building of syntactic trees) based on the assemblies operations project and merge and involving the medial temporal lobe, the superior temporal gyrus, and Broca's area of the left human brain.

References

1 Rosa I. Arriaga, David Rutter, Maya Cakmak, and Santosh S. Vempala. Visual categorization with random projection. *Neural Computation*, 27(10):2132–2147, 2015. URL: `http://dx.doi.org/10.1162/NECO_a_00769`, `doi:10.1162/NECO_a_00769`.

2 Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006. URL: `http://dx.doi.org/10.1007/s10994-006-6265-7`, `doi:10.1007/s10994-006-6265-7`.

3 M.F. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65(1):79–94, 2006.

4 Robert C Berwick and Noam Chomsky. *Why only us: Language and evolution.* MIT Press, 2016.

5 G Buzsaki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3), 2010.

6 S. DasGupta. Learning mixtures of gaussians. In *Proc. of FOCS*, 1999.

7 Sanjoy Dasgupta, Charles F. Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, 2017. URL: `http://science.sciencemag.org/content/358/6364/793`, `arXiv:http://science.sciencemag.org/content/358/6364/793.full.pdf`, `doi:10.1126/science.aam9868`.

8 Emanuela De Falco, Matias J Ison, Itzhak Fried, and Rodrigo Quian Quiroga. Long-term coding of personal and universal associations underlying the memory web in the human brain. *Nature Communications*, 7:13408, 2016.

9 Nai Ding, Lucia Melloni, Hang Zhang, Xing Tian, and David Poeppel. Cortical tracking of hierarchical linguistic structures in connected speech. *Nature neuroscience*, 19(1):158, 2016.

10 S. M. Frankland and J. D. Greene. An architecture for encoding sentence meaning in left mid-superior temporal cortex. *Proceedings of the National Academy of Sciences*, 112(37):11732–11737, 2015.

11 Kevin M Franks, Marco J Russo, Dara L Sosulski, Abigail A Mulligan, Steven A Siegelbaum, and Richard Axel. Recurrent circuitry dynamically shapes the activation of piriform cortex. *Neuron*, 72(1):49–56, 2011.

12 Kenneth D Harris. Neural signatures of cell assembly organization. *Nature Reviews Neuroscience*, 6(5):399, 2005.

13 Kenneth D Harris, Jozsef Csicsvari, Hajime Hirase, George Dragoi, and György Buzsáki. Organization of cell assemblies in the hippocampus. *Nature*, 424(6948):552, 2003.

14 Matias J Ison, Rodrigo Quian Quiroga, and Itzhak Fried. Rapid encoding of new memories by individual neurons in the human brain. *Neuron*, 87(1):220–230, 2015.

15 Sheena A Josselyn, Stefan Köhler, and Paul W Frankland. Finding the engram. *Nature Reviews Neuroscience*, 16(9):521, 2015.

16 R. Legenstein, W. Maass, C. H. Papadimitriou, and S. S. Vempala. Long-term memory and the densest k-subgraph problem. In *Proc. of 9th Innovations in Theoretical Computer Science (ITCS) conference, Cambridge, USA, Jan 11-14. 2018*, 2018.

17 Robert Legenstein, Christos H Papadimitriou, Santosh Vempala, and Wolfgang Maass. Assembly pointers for variable binding in networks of spiking neurons. *arXiv preprint arXiv:1611.03698*, 2016.

18 Eva Pastalkova, Vladimir Itskov, Asohan Amarasingham, and György Buzsáki. Internally generated cell assembly sequences in the rat hippocampus. *Science*, 321(5894):1322–1327, 2008.

19 C. Pokorny, M. J. Ison, A. Rao, R. Legenstein, C. Papadimitriou, and W. Maass. Associations between memory traces emerge in a generic neural circuit model through STDP. *bioRxiv:188938*, 2017.

20 Leslie G. Valiant. *Circuits of the mind.* Oxford University Press, 1994.

21 Leslie G. Valiant. A neuroidal architecture for cognitive computation. *J. ACM*, 47(5):854–882, 2000. `doi:10.1145/355483.355486`.

22 Santosh Srinivas Vempala. *The Random Projection Method*, volume 65 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science.* DIMACS/AMS, 2004. URL: `http://dimacs.rutgers.edu/Volumes/Vol65.html`.

23 Emiliano Zaccarella and Angela D. Friederici. Merge in the human brain: A sub-region based functional investigation in the left pars opercularis. *Frontiers in Psychology*, 6:1818, 2015. URL: `https://www.frontiersin.org/article/10.3389/fpsyg.2015.01818`, `doi:10.3389/fpsyg.2015.01818`.

24 Emiliano Zaccarella, Lars Meyer, Michiru Makuuchi, and Angela D Friederici. Building by syntax: the neural basis of minimal linguistic structures. *Cerebral Cortex*, 27(1):411–421, 2017.