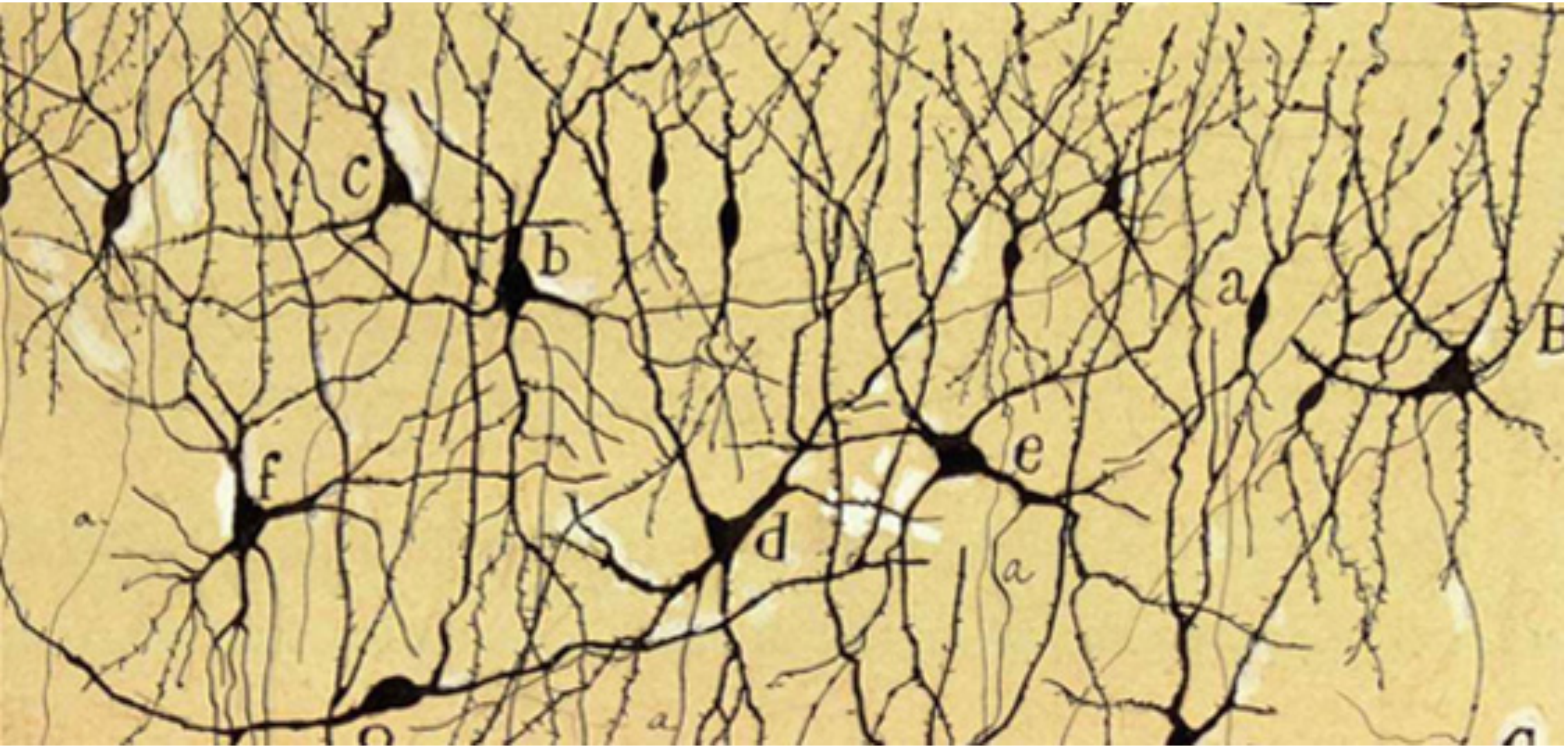
What Should a Computational Theory of Cortex Explain?

Leslie Valiant
Harvard University



Ramón y Cajal

Advice for a Young Investigator

by Ramón y Cajal

Chapter 5: Diseases of the Will

Contemplators.

Bibliophiles and polyglots.

Megalomaniacs.

Instrument addicts.

Misfits.

Theorists.

“Let us emphasize again this obvious conclusion: a scholar’s positive contribution is measured by the sum of the original data that he contributes. Hypotheses come and go but data remain. Theories desert us, while data defend us. They are our true resources, our real estate, and our best pedigree. In the eternal shifting of things, only they will save us from the ravages of time and from the forgetfulness or injustice of men. To risk everything on the success of one idea is to forget that every fifteen or twenty years theories are replaced or revised. So many apparently conclusive theories in physics, chemistry, geology, and biology have collapsed in the last few decades! On the other hand, the well-established facts of anatomy and physiology and of chemistry and geology, and the laws and equations of astronomy and physics remain—immutable and defying criticism.”

Ramón y Cajal (1894?)

Computation and the Brain

- “We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions ... .” Turing (1936).
- “It is easily shown … that every net …can compute only such numbers as can a Turing machine; that each of the latter numbers can be computed by such a net … . This is of interest as affording psychological justification of the Turing definition of computability … .” McCulloch and Pitts (1943).

.

.

.

- 1950s onward: “It is computation, stupid.”

David Marr (1982):

1. *Computational theory*: What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?
2. *Representation and algorithm*: How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?
3. *Hardware implementation*: How can the representation and algorithm be realized physically?

Marr: Theories of Cerebellum (1969), Cortex (1970), Hippocampus (1971).

Marr (1973): “I do not expect to write many more papers in theoretical neurophysiology – at least not for a long time: but … I shall be very surprised if my 1969 or 1971 papers turn out to be very wrong.”

Marr (1975): “With problems of biological information processing there has been almost no experience, and one's intuition is at best untrustworthy. It may even be that biological information processing admits of no general theories except ones so unspecific as to have only descriptive, and no predictive powers.”

David Marr (1982):

1. *Computational theory*: What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?
2. *Representation and algorithm*: How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?
3. *Hardware implementation*: How can the representation and algorithm be realized physically?

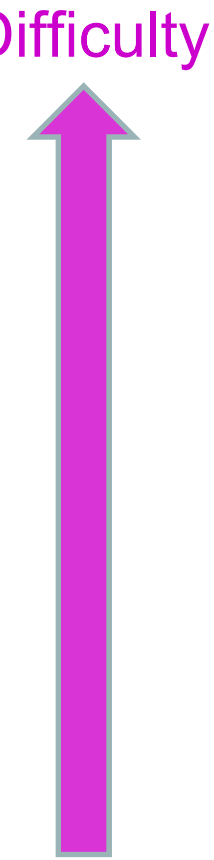
David Marr (1982):

1. *Computational theory*: What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?
2. *Representation and algorithm*: How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?
3. *Hardware implementation*: How can the representation and algorithm be realized physically?

4. Marr +

   Model of computation with costs +

   Explanations of quantitatively challenging tasks



1. Communication of n-bit sequence.

Difficulty

2. Computation of n-argument function.

1. Communication of n-bit sequence.

Difficulty

3. Inductive learning of *n*-argument function.

2. Computation of *n*-argument function.

1. Communication of *n*-bit sequence.

4. Darwinian evolution of *n*-argument function.

Difficulty

3. Inductive learning of *n*-argument function.

2. Computation of *n*-argument function.

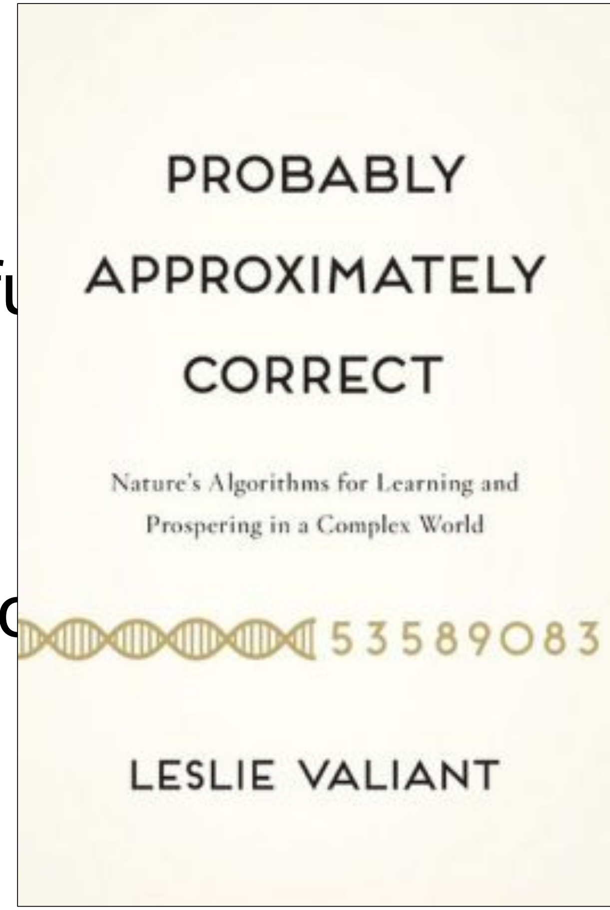
1. Communication of *n*-bit sequence.

4. Darwinian evolution of *n*-argument function.

3. Inductive learning of *n*-argument fu

2. Computation of *n*-argument functio

1. Communication of *n*-bit sequence.



4. Darwinian evolution of *n*-argument function.

Difficulty

3. Inductive learning of *n*-argument function.

2. Computation of *n*-argument function.

1. Communication of *n*-bit sequence.

4. Darwinian evolution of *n*-argument function.

Random jumble of neurons should do something useful.

Difficulty

3. Inductive learning of *n*-argument function.

2. Computation of *n*-argument function.

1. Communication of *n*-bit sequence.

4. Darwinian evolution of *n*-argument function.

Random jumble of neurons should do something useful.

Difficulty

3. Inductive learning of *n*-argument function.

Perceptrons, Nearest Neighbors, are powerful.

2. Computation of *n*-argument function.

1. Communication of *n*-bit sequence.

Given a Model of Computation:

REPRESENTATION
Data Structure

ALGORITHM

TASK SPECIFICATION

Marr (1975): "*... the primary unresolved issue is what functions you want implemented, and why.*"

The Brain is Communication-challenged

Each neuron connected to a small fraction of the others, and the influences of many connections are small …..

Random Access Tasks: Type I

Allocate Storage to New Concept

Random Access Tasks: Type I
Allocate Storage to New Concept

(1) e.g. First time you heard of "Tiger Woods".

Random Access Tasks: Type I
Allocate Storage to New Concept

(1) e.g. First time you heard of "Tiger Woods".

Hierarchical Memorization: For any stored items A, B, allocate neurons to new item C and change synaptic weights so that in future A and B active will cause C to be active also.

Random Access Tasks: Type I
Allocate Storage to New Concept

(1) e.g. First time you heard of "Tiger Woods".

Hierarchical Memorization: For any stored items A, B, allocate neurons to new item C and change synaptic weights so that in future A and B active will cause C to be active also. (Chunking)

Random Access Tasks: Type II

Add relationships among represented concepts

Random Access Tasks: Type II

Add relationships among represented concepts

(2) (e.g. David Beckham → Retired)

Association: For any stored items A, B, change synaptic weights so that in future when A is active then B will be caused to be also.

Random Access Tasks: Type II

Add relationships among represented concepts

(2) (e.g. David Beckham → Retired)

Association: For any stored items A, B, change synaptic weights so that in future when A is active then B will be caused to be also.

(3) Inductive Learning: of simple threshold functions from examples.

(4) Supervised Memorization of conjunctions: For items A, B, C want that if in future A and B activated then C activated also.

WHERE THERE ARE NO VIABLE THEORIES:

1. Use neural model that *underestimates* the brain.
2. Specify some challenging set of *multiple* task types.
3. Show that these task types *can* be executed on the model.
4. **Show that *sequences of thousands of interleaved tasks of these types* can be supported without degradation of earliest.**

The Neuroidal Model of Computation

Resource Bounds: n neurons, each connected to d others, with max. synaptic weights $1/k \times$ threshold.

Local Updates to states and synapses …..

Timing: Inputs to neurons can be activated simultaneously by *prompt*. Neurons have local timing mechanism which can keep them in sync. for a few steps.

Representations

Each real world "item" corresponds to a set of (about r) neurons.

Note: Correspondences between items and neurons are "experimentally determinable"

r is large.

(e.g. in hippocampus, IT, olfactory bulb)

Representations: Disjoint or Shared?

- Disjoint: Each neuron represents just one, possibly complex, item.
- Shared: Each neuron may represent many items.

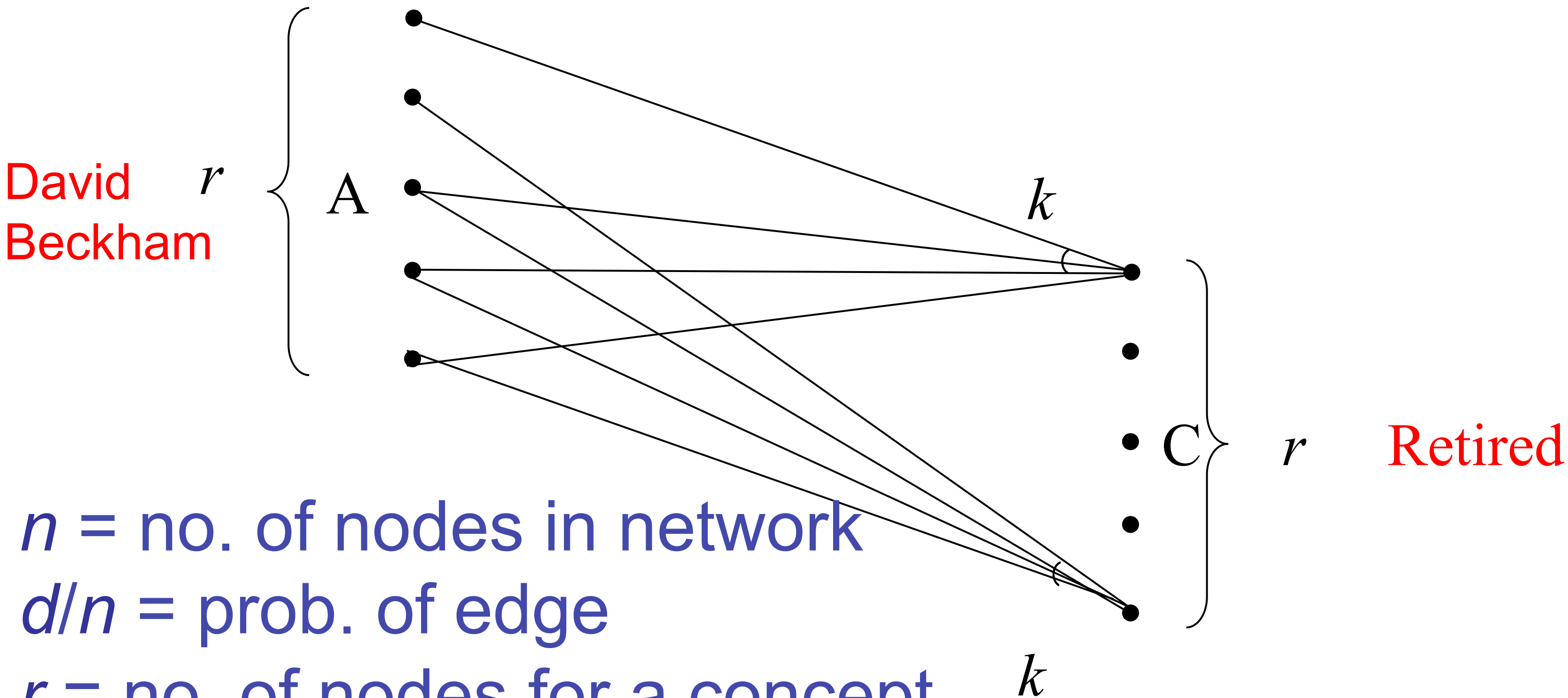
Random Graphs

Expanders: For n node graphs, for some constants b, $c > 0$, every set of bn nodes have at least $(b+c)n$ neighbors in total.

Fact: Random graphs are expanders.

Kolmogorov and Barzdin (1967): Fixed degree graphs on n nodes that are "expanders" need volume $n^{3/2}$.

Network Requirement for Association



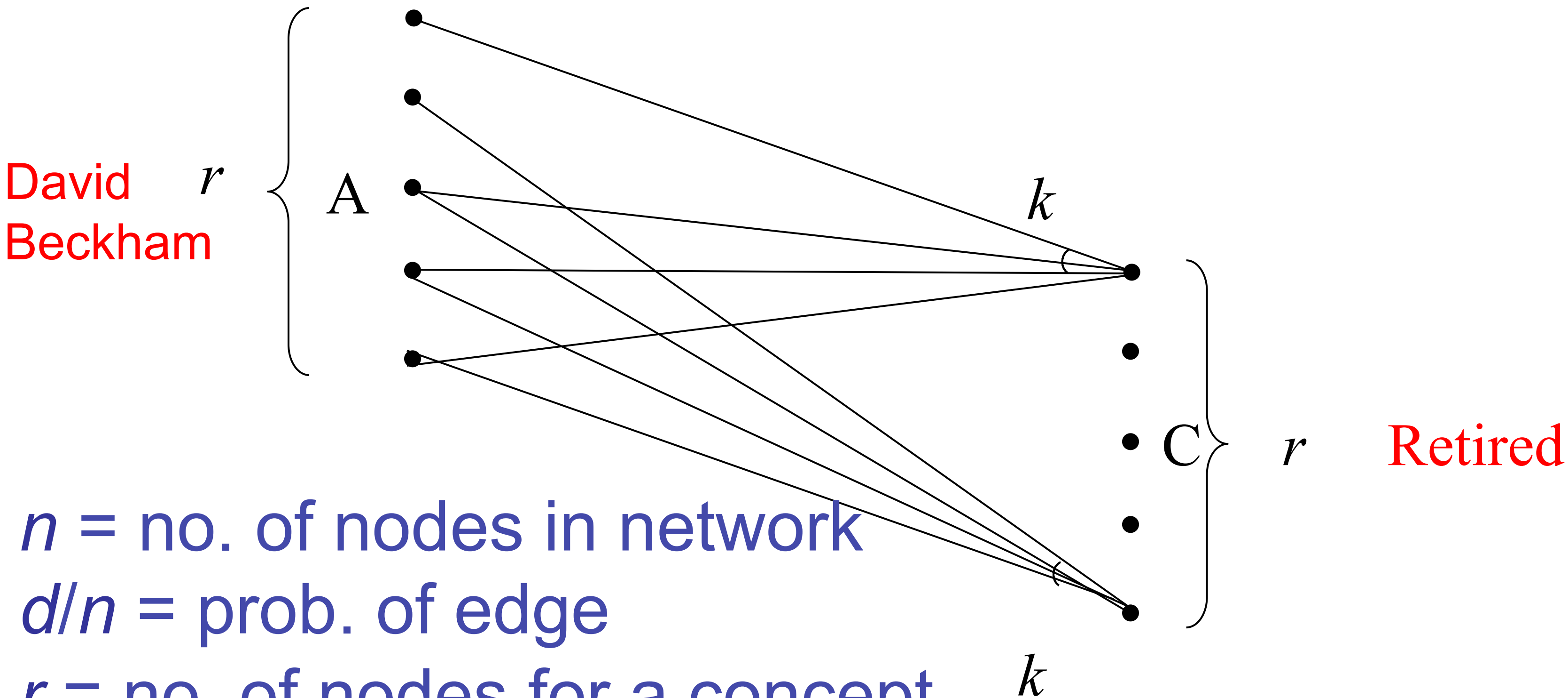
n = no. of nodes in network

d/n = prob. of edge

r = no. of nodes for a concept

k = min. no. of inputs to fire a node

Network Requirement for Association

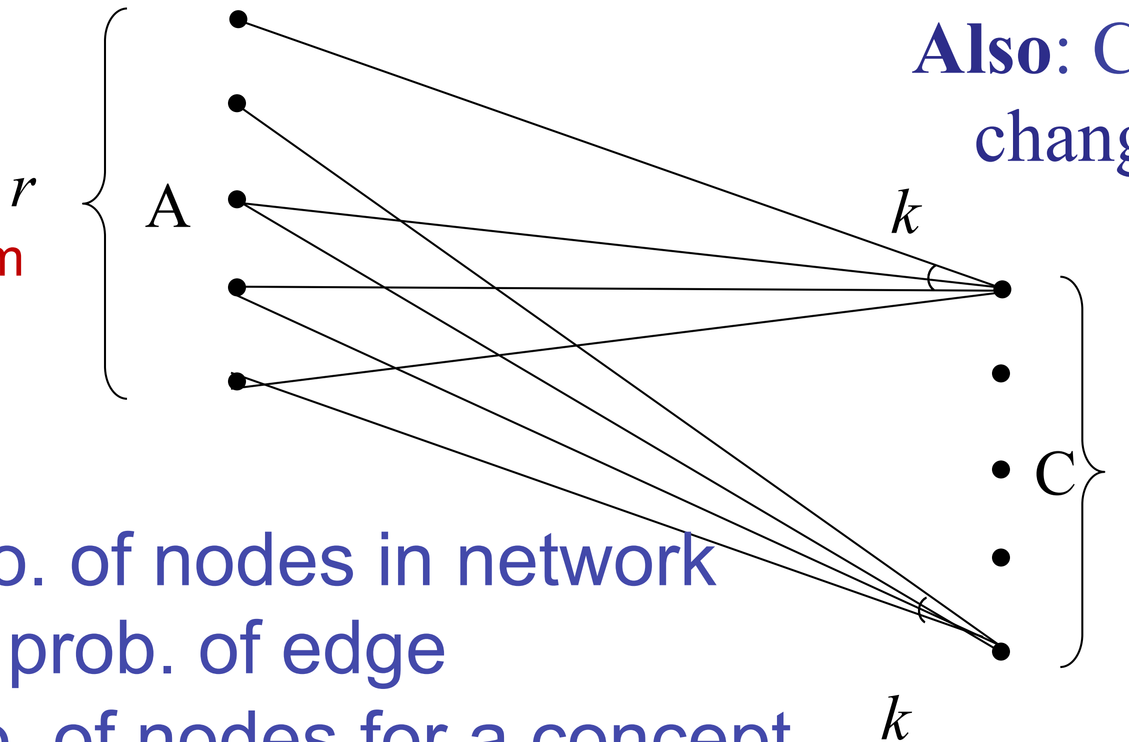


n = no. of nodes in network
d/n = prob. of edge
r = no. of nodes for a concept
k = min. no. of inputs to fire a node

$$\beta(r, d/n, k) \sim 1 \text{ if } rd/n >\sim k.$$

Network Requirement for Association

David Beckham



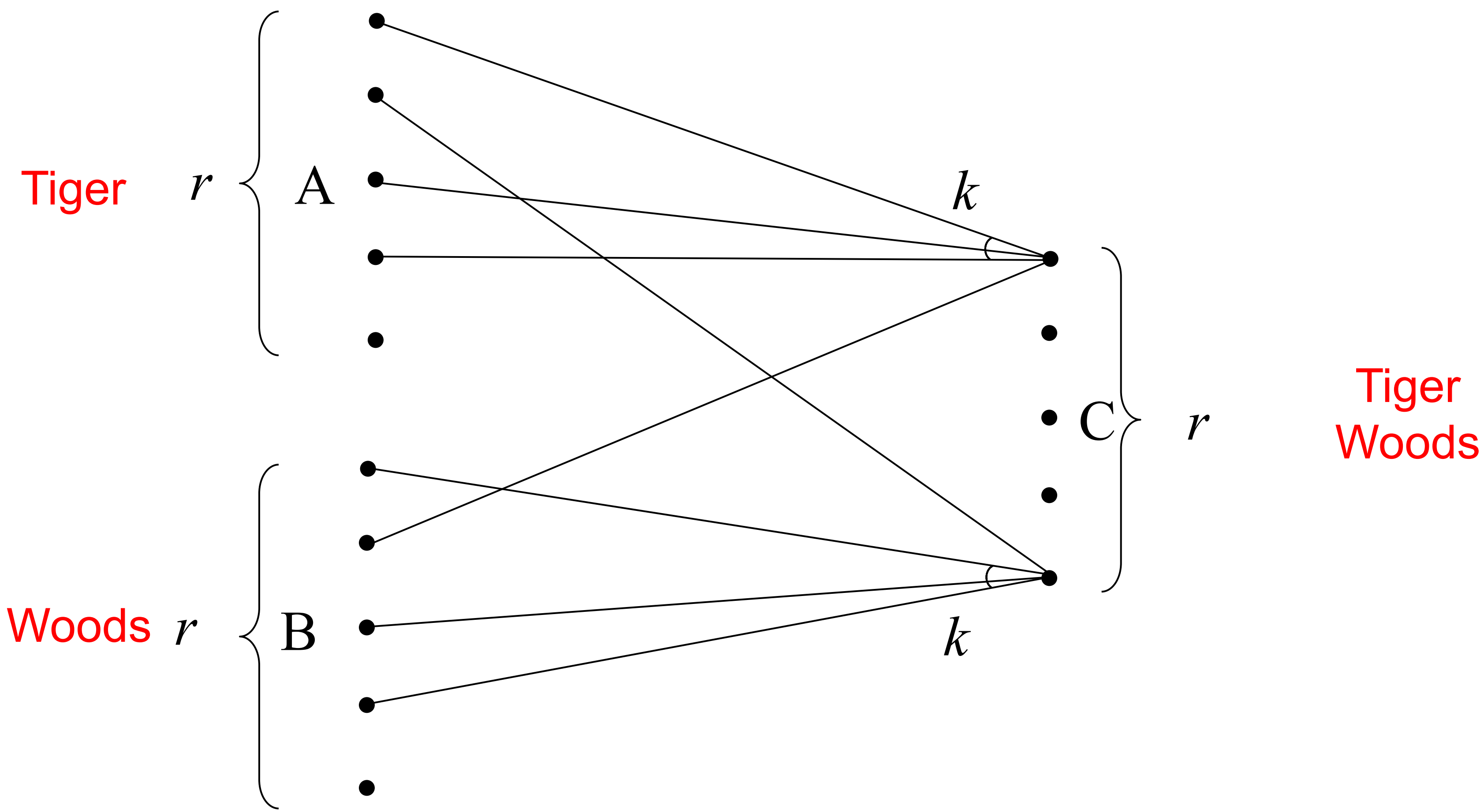
Also: Can realize weight changes on neuroidal model.

Retired

n = no. of nodes in network
d/n = prob. of edge
r = no. of nodes for a concept
k = min. no. of inputs to fire a node

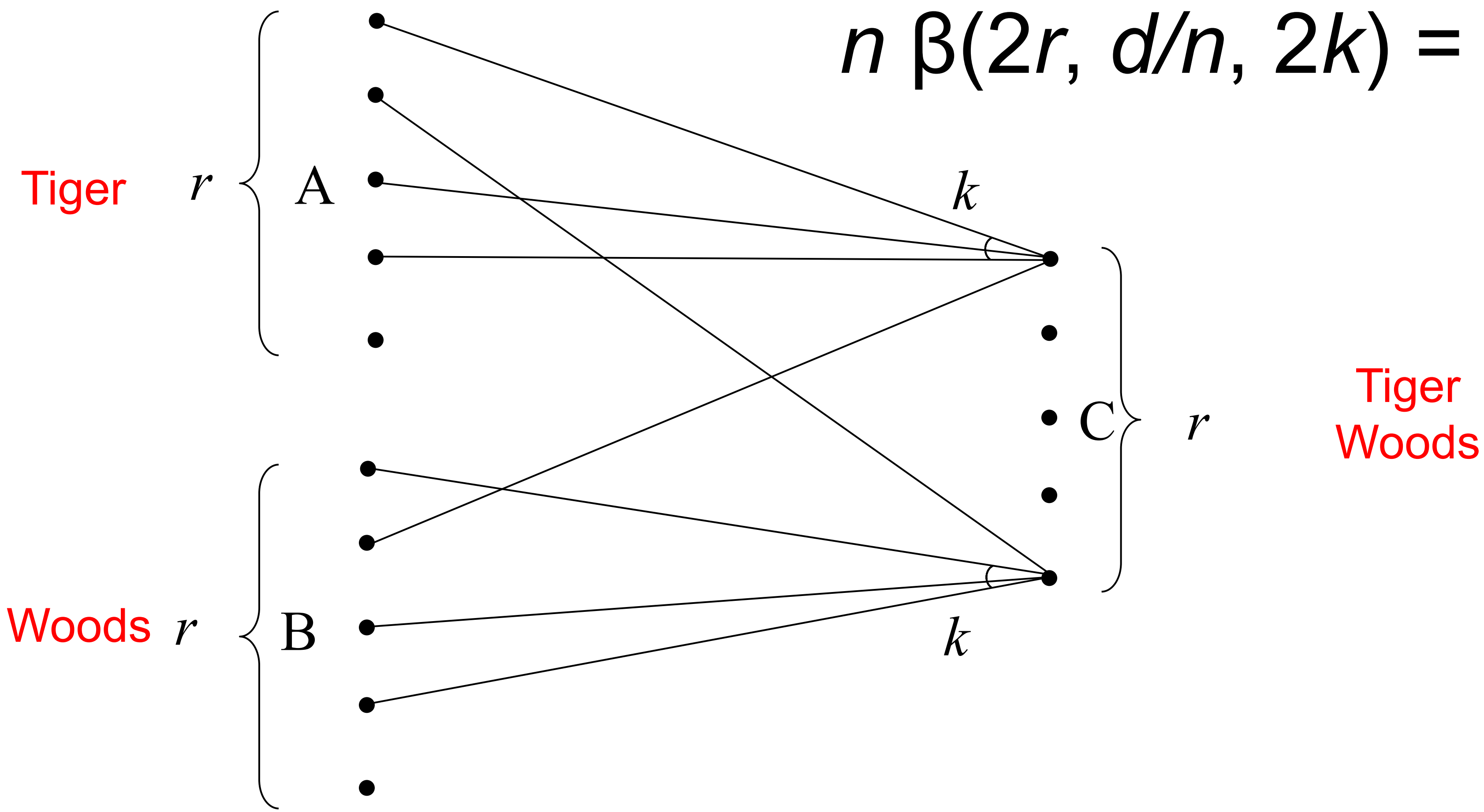
$$\beta(r, d/n, k) \sim 1 \text{ if } rd/n >\sim k.$$

Hierarchical Memorization



Hierarchical Memorization

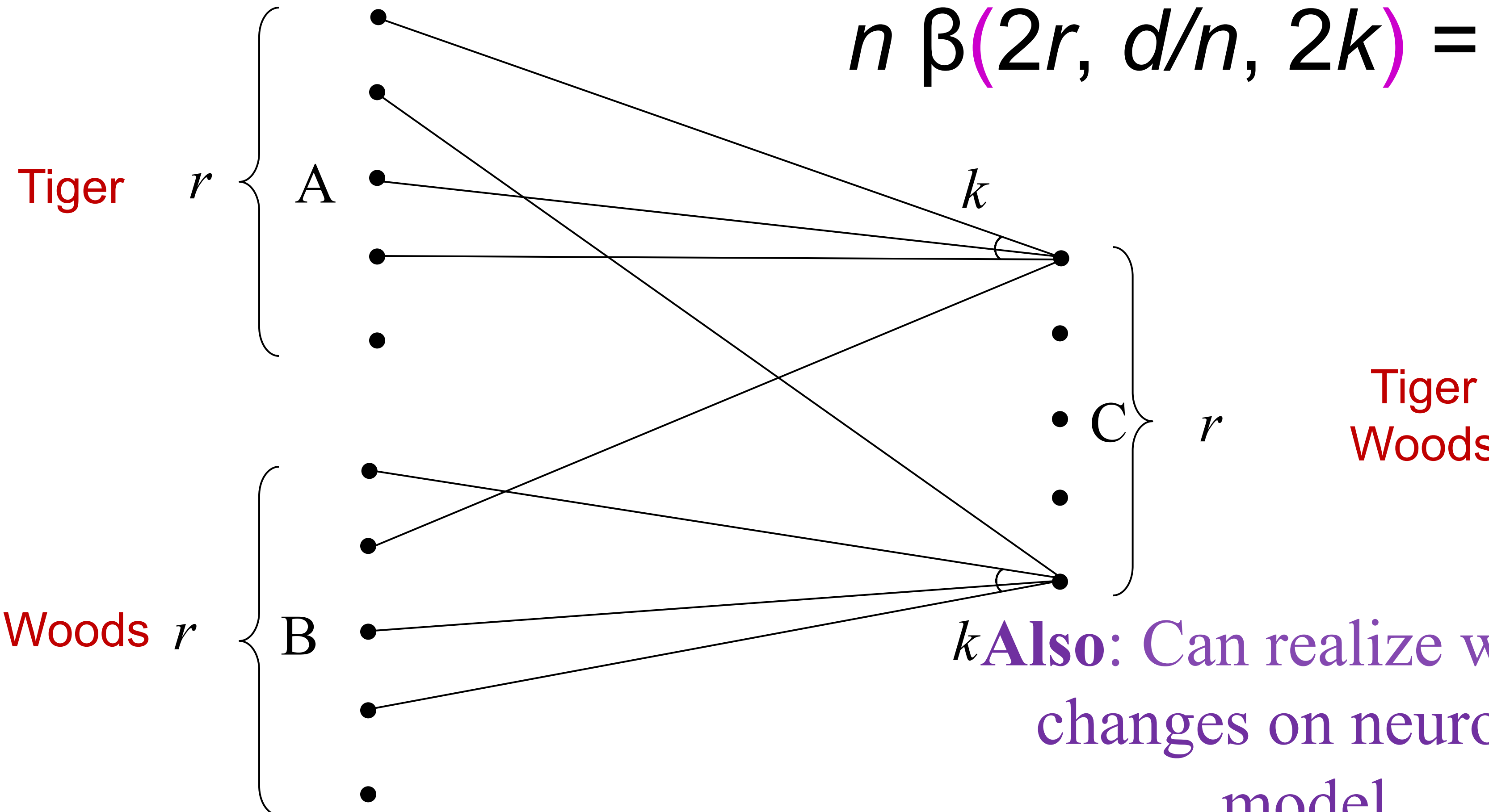
$$n\ \beta(2r, d/n, 2k) = r$$



$$n\ \beta(r, d/n, k)^2 = r$$

Hierarchical Memorization

$$n\,\beta(2r,\ d/n,\ 2k) = r$$



Also: Can realize weight changes on neuroidal model.

$$n\,\beta(r,\ d/n,\ k)^2 = r$$

Actual Conditions Are More Complex

e.g. for Hierarchical Memorization:

A and B C

(1) Expected number of nodes suitably connected to both A and B is *r*.

(2) If fewer than “a half” of A fire and all of B then fewer than “a half” of C will fire.

(3) If a different conjunction, say A and D, fire then less than a half of C fire.

Some Large Scale Simulations

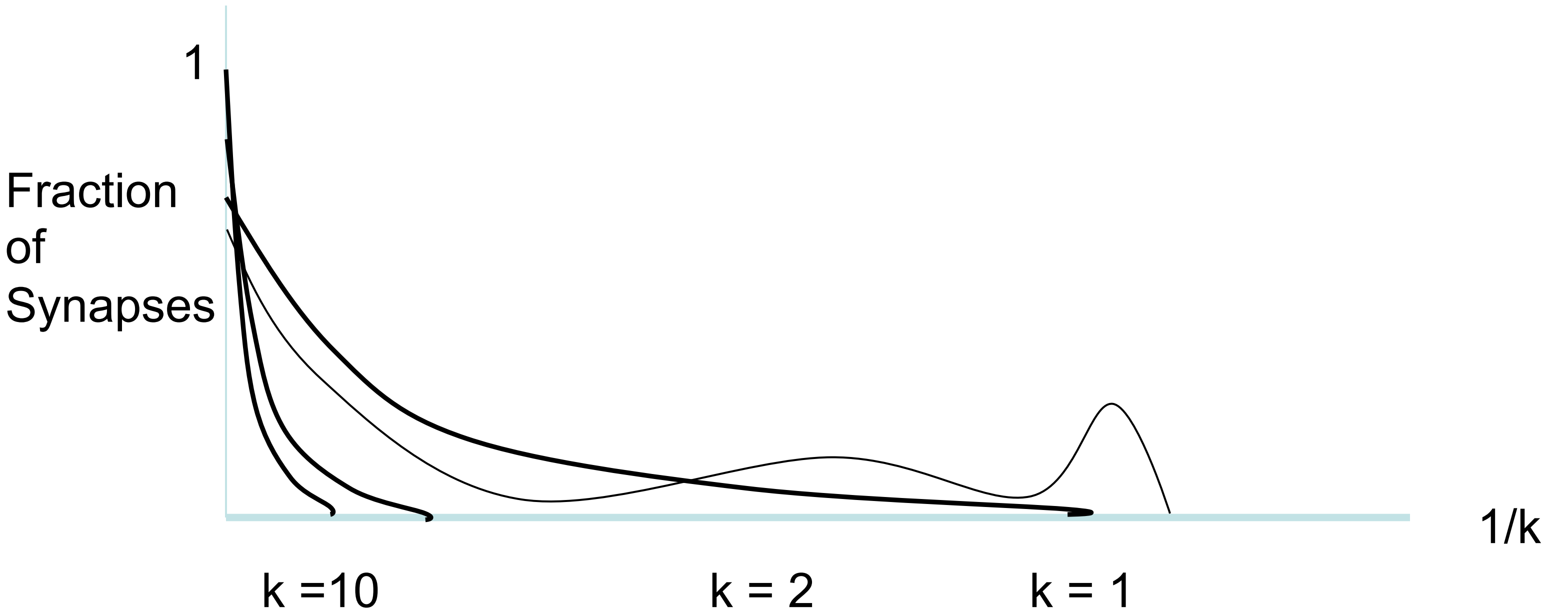
(V. Feldman & LV, Neural Computation, 2009)

Simulate mixed sequences of associations, supervised memorization, and inductive learning tasks, on initial allocation by hierarchical memorization.

What is capacity, the total number of tasks before effectiveness of acquired functions (even those done early) degrades?

A Most Basic Physiological Question

What is effective strength of influence of cortical neurons on each other? (via dynamic synapses.)



Thesis : The Two Regimes

- **Regime [V04] :** Weak synapses, $8 \leq k < 1000$:
  Shared dense representations,
  (Large r, Item-neuron correspondences easier to identify)
  (? IT, hippocampus, olfactory systems?)

- **Regime [V94]:** Strong synapses, $k = 1$:
  Disjoint sparse representations,
  (Small r, Item-neuron correspondences hard to identify)
  (? Role in prefrontal cortex?)

Results of Simulations: Regime α

(V.Feldman & LV, Neural Computation, 2009)

$n = 10^8$ neurons.

$d = 8{,}000$ connections per neuron.

$k = 16$ (i.e. inputs from 16 needed for a.p.)

$r = 360000$ neurons per item, shared.

$\downarrow$

Sequences of 3,200 actions can be supported with small interference.

Results of Simulations: Regime β

(V.Feldman & LV, Neural Computation, 2009)

$n = 10^8$ neurons.

$d = 4{,}000$ connections per neuron.

$k = 1$ (i.e. maximally strong synapses)

$r = 100$ neurons per item, disjoint.

$\downarrow$

Sequences of 250,000 actions can be supported with small interference.

Thesis : The Two Regimes

- **Regime [illegible] [V04] :** Weak synapses, $8 \leq k < 1000$:
  Shared dense representations,
  (Large r, Item-neuron correspondences easier to identify)
  (? IT, hippocampus, olfactory systems?)
  Moderate capacities 10^3 -10^4
  Very simple algorithms – first to evolve?
- **Regime [illegible] [V94]:** Strong synapses, $k = 1$:
  Disjoint sparse representations,
  (Small r, Item-neuron correspondences hard to identify)
  (? Role in prefrontal cortex?)
  Large capacities $>10^5$
  Slightly less simple algorithms – harder to evolve - ???

Locust

Rule of Thumb: $k \sim rd/n$

For weak synapses parameters satisfy:

$$\beta(r, d/n, k)^2 = r/n$$

Expected value of r trials is rd/n. Hence

$$\text{Prob}(> \text{k successes}) = \text{small}$$

when $k >\sim rd/n$.

n = no. of nodes in network
d/n = prob. of edge
r = no. of nodes for a concept
k = no. of inputs to fire a node

Olfactory System of Locust

(Jortner, Farivar and Laurent, 2007)

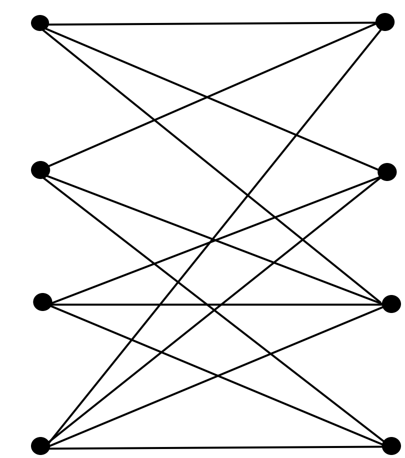
Bipartite graph: projection neurons KENYON CELLS

$n = 830$ inputs, $r = 100 - 150.$

$N = 50{,}000$ outputs, $R = 5 - 250.$

$D/n = .5 \pm .13,$

$K = 100.$



p.n. K.C.

Theory predicts : $K >\sim rD/n.$

More exactly: $\beta(r, D/n, K) = R/N.$

e.g. fit with: $\beta(138, .63, 100) = .0015.$

A Stability Problem

Fact: Hierarchical Memorization mechanism is unstable [V94, Gerbessiotis 03].

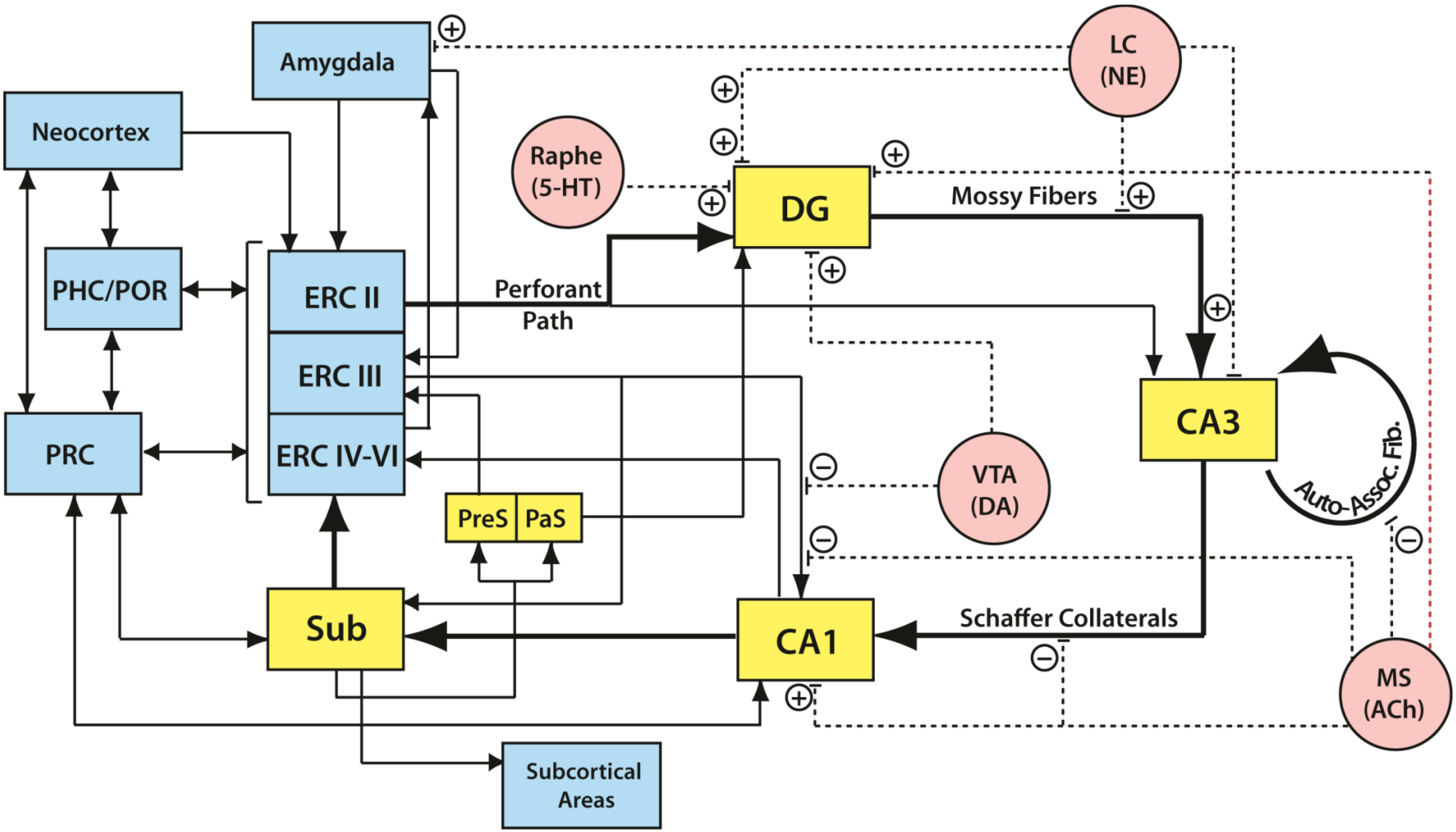
The simulations for 250,000 acts were for one level of memory allocation, but arbitrary everything else.

Solution 1: Allocate to fixed depth (naming), then build arbitrary data structure via other operations [V94].

Solution 2: Other cortical – Gunay & Maida [06], Beal & Knight [08].

Solution 3 [V12]**:** The hippocampus does hierarchical memory allocation. (c.f. Wickelgren [79] chunking.)

Hippocampus (in yellow)



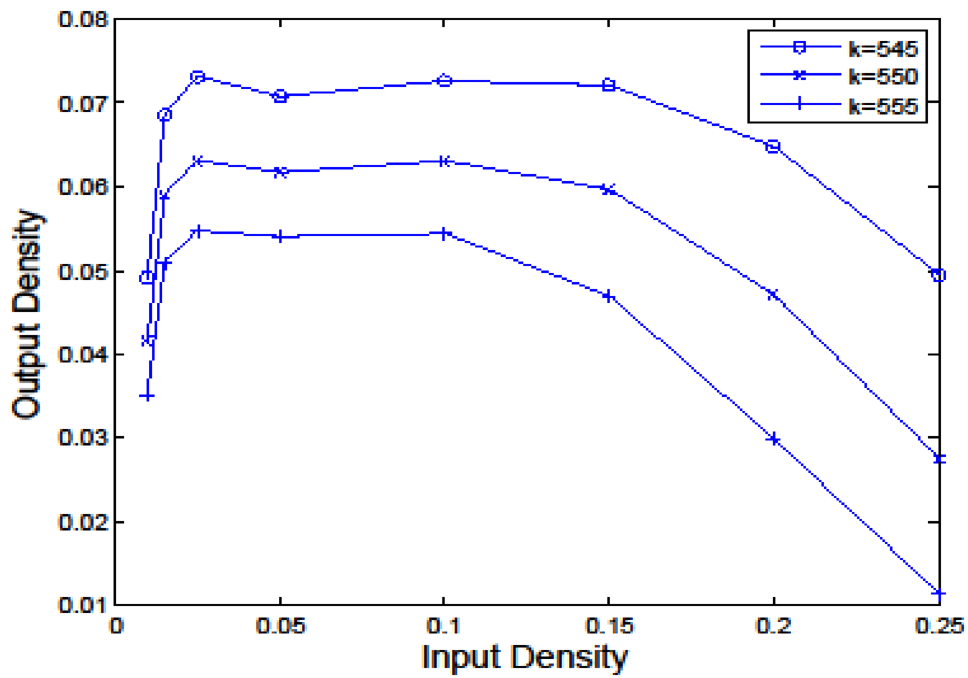
pbs.jhu.edu

1. View the hippocampus as a feedforward network of say 2-4 layers, each layer a random bipartite graph, each neuron a threshold element.

2. Analyze what this does for *stability of number of neurons allocated*, for different activity densities, weights.

The Hippocampus as Stable Memory Allocator for Cortex

Input	Level 1	Level 2	Level 3	Level 4
0.0400	0.00135	0.00348	0.00713	0.00974
0.0300	0.00315	0.00667	0.00958	0.00997
0.0250	0.00464	0.00834	0.00996	0.00994
0.0200	0.00650	0.00950	0.00997	0.00993
0.0150	0.00854	0.00996	0.00995	0.00993
0.0100	0.00992	0.00995	0.00995	0.00993
0.0075	0.00983	0.00996	0.00992	0.00993
0.0050	0.00865	0.01000	0.00992	0.00995
0.0033	0.00690	0.00967	0.00996	0.00994
0.0020	0.00482	0.00849	0.00996	0.00993
0.0015	0.00383	0.00754	0.00984	0.00994
0.0010	0.00271	0.00603	0.00929	0.00999



k=545
k=550
k=555
Output Density
Input Density
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.05
0.1
0.15
0.2
0.25

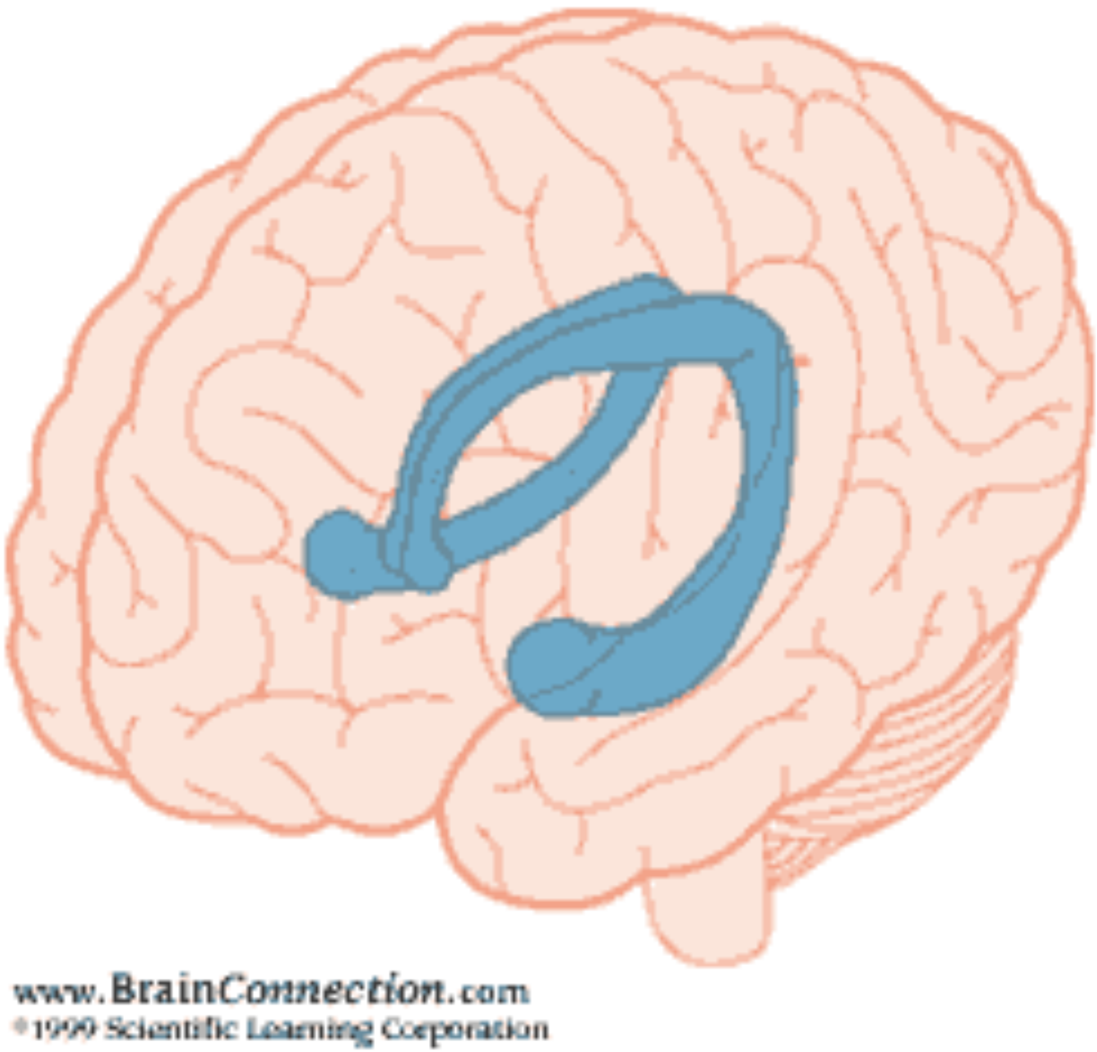
Stable Memory Allocator

Like a hash function, with properties that:

1. Stable
2. Continuous
3. Orthogonal

To show that networks of threshold gates with plausible parameters have these three properties.

Theory: Hierarchical memory allocation in cortex essential. Mediated by hippocampus which ensures stability of numbers of neurons allocated to various concepts.



The Choices

Model of Computation: neuroidal

Representation: positive

Algorithms: vicinal

Tasks: Creating circuits for 4 tasks of learning or memorization (versus storing strings.)

THANK YOU