# The CEREBELLUM as NEURAL ASSOCIATIVE MEMORY

Pentti Kanerva
<pkanerva@berkeley.edu>
Redwood Center for Theoretical Neuroscience

. Cerebellum figures and facts
. The cerebellum challenge
. Development of theory from top down
. Representing concepts with high-dim. vectors
. Computing with high-D vectors
. The necessity for an associative memory
. Cerebellum as a model from nature

# Cerebellum Figures and Facts

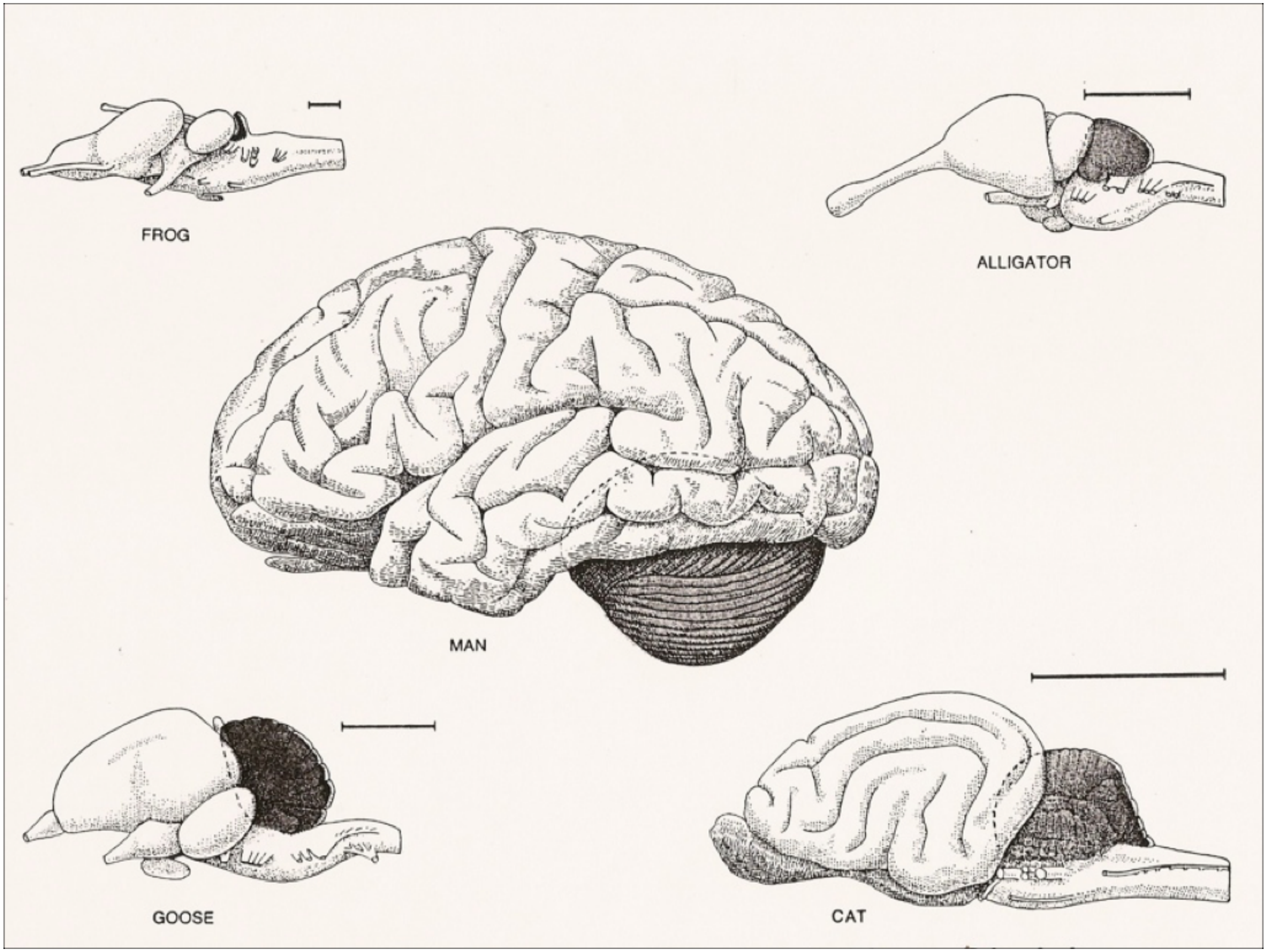


FIGURE 1. Cerebellum as part of vertebrate brains.

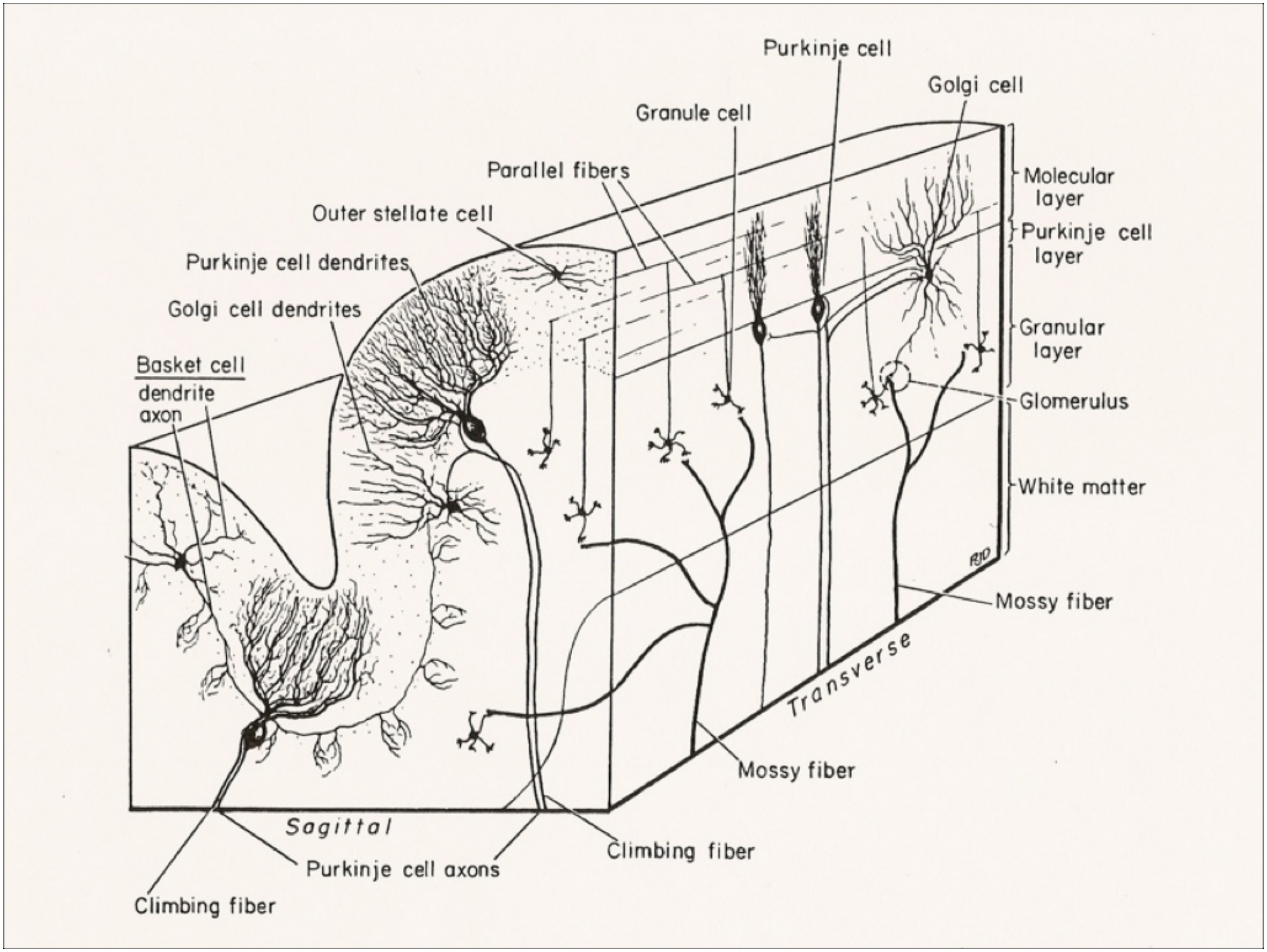


FIGURE 2. Cerebellum cell types in 3D.

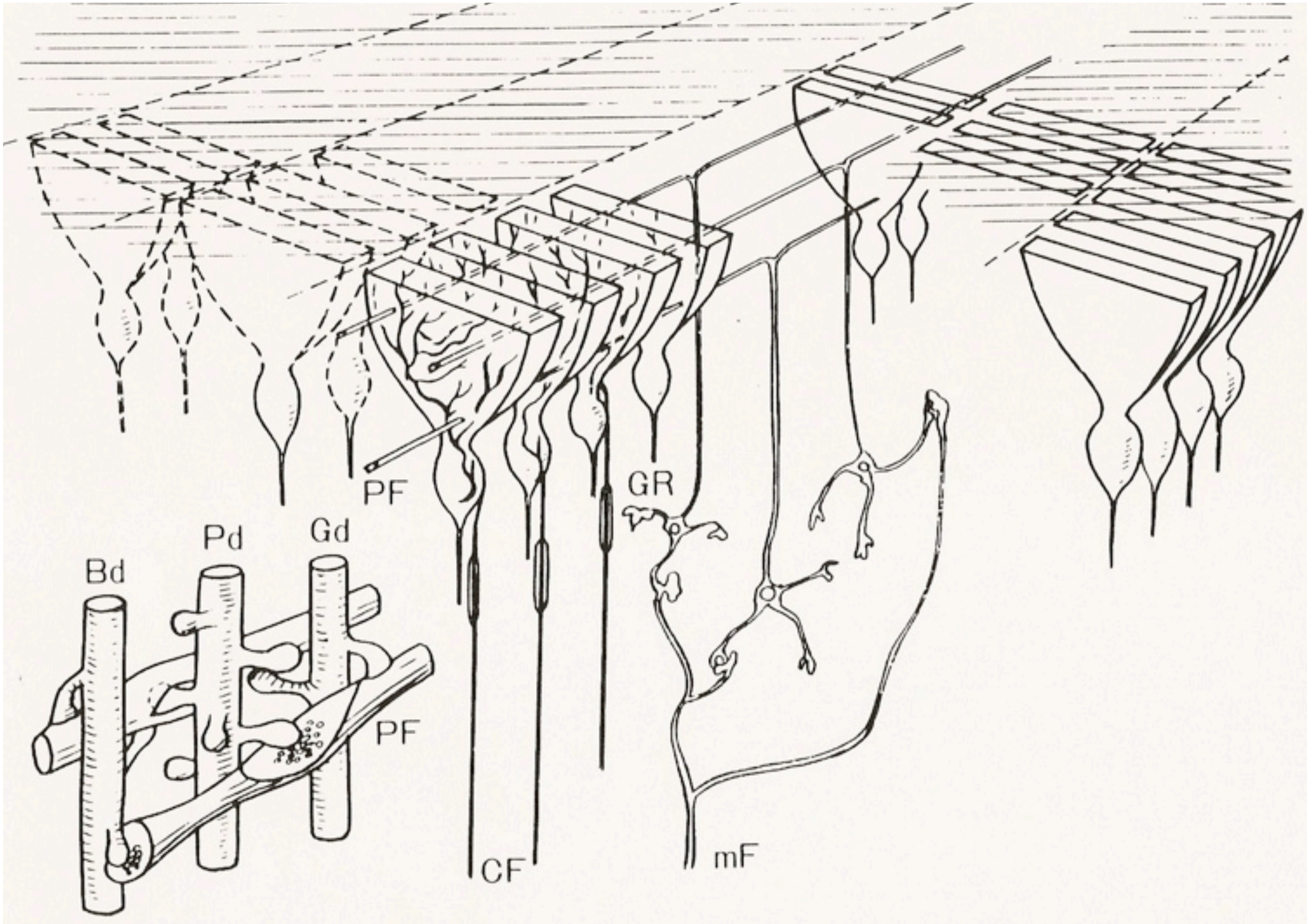


**FIGURE 3. 3D organization of the "main" circuit.**

**Cerebellum Facts and Figures**

. 200 million Mossy Fibers: input from outside

. 40 billion Granule Cells, the most numerous

. 15 million Purkinje Cells (PC): sole output

. Climbing Fibers: input from within
-- 1/PC, shared by approx. 10 PCs

100,000 synapses/PC
. 1.5 trillion synapses overall

**How big is 1.5 trillion?**

1.5 trillion synapses @ 1 bit/synapse
= 360,000 books, 400 pages each
= 4 miles of shelf space

## The Cerebellum Challenge

Traditional theories--logic, rule-based AI, artificial neural nets, connectionism, parallel distributed processing, deep learning--*leave too much unexplained and unexplored.*

For example, **why the cerebellum?**

. It has 40 billion neurons
vs. 16 billion in the rest of the brain

. Its organization is simple and regular

*The cerebellum must fulfill some essential function that computational theories and models of the brain cannot afford to ignore*

**What Is the Cerebellum for?**

. Coordination of movement, fine *motor control*

-- learn, generate, and monitor sequences

-- predict

. Growing evidence for *higher cognitive functions* such as language

-- this agrees with the theme of this talk:

*Theory of computing with high-dimensional vectors assumes a high-capacity associative memory*

## Development of Theory from Top Down

Top-down development prepares the mind to *recognize an answer* when it presents itself

1. **Philosophy and Psychology**
   . The character of concepts

2. **Mathematics**
   . Develop a mathematical model of the world of concepts

3. **Engineering**
   . "Build" a physical structure implied by the model

4. **Biology**
   . Is there anything like it in the brain?
   . Is anything of essence missing?

## Representing Concepts with High-D Vectors

**Brains** consist of *neurons* but **minds** work with *concepts*

The world of concepts is
- . huge and
- . ever-expanding

Representation of concepts must allow for that

**Concepts** can be compared for *similarity of meaning*

man ≈ woman
man ≉ lake

**Distant concepts** have *similar neighbors*

man ≉ lake
man ≈ fisherman ≈ fish ≈ lake
man ≈ plumber ≈ water ≈ lake
plumber ≉ fish

**Robustness of Concepts** (and of percepts)

. Sensory input never repeats exactly
  -- yet we recognize people and things
. Recognition is fast and extremely tolerant of
  variation and "noise"
. Learning can be very fast
  -- from a single exposer
  -- a handful of examples
  -- explicit instruction
. Memories can last a lifetime

How to model the world of concepts?

*What mathematical objects would have the above properties?*

**Properties of High-Dimensional Vectors**

e.g. 10,000-dimensional binary vectors

. 10K-bit vectors/"words"/points of a 10K-dim. space

. Total number of 10K-bit vectors: $2^{10,000}$

. Hamming distance *H* between 10K-bit vectors follows the binomial distribution:
-- mean = 5,000, STD = 50

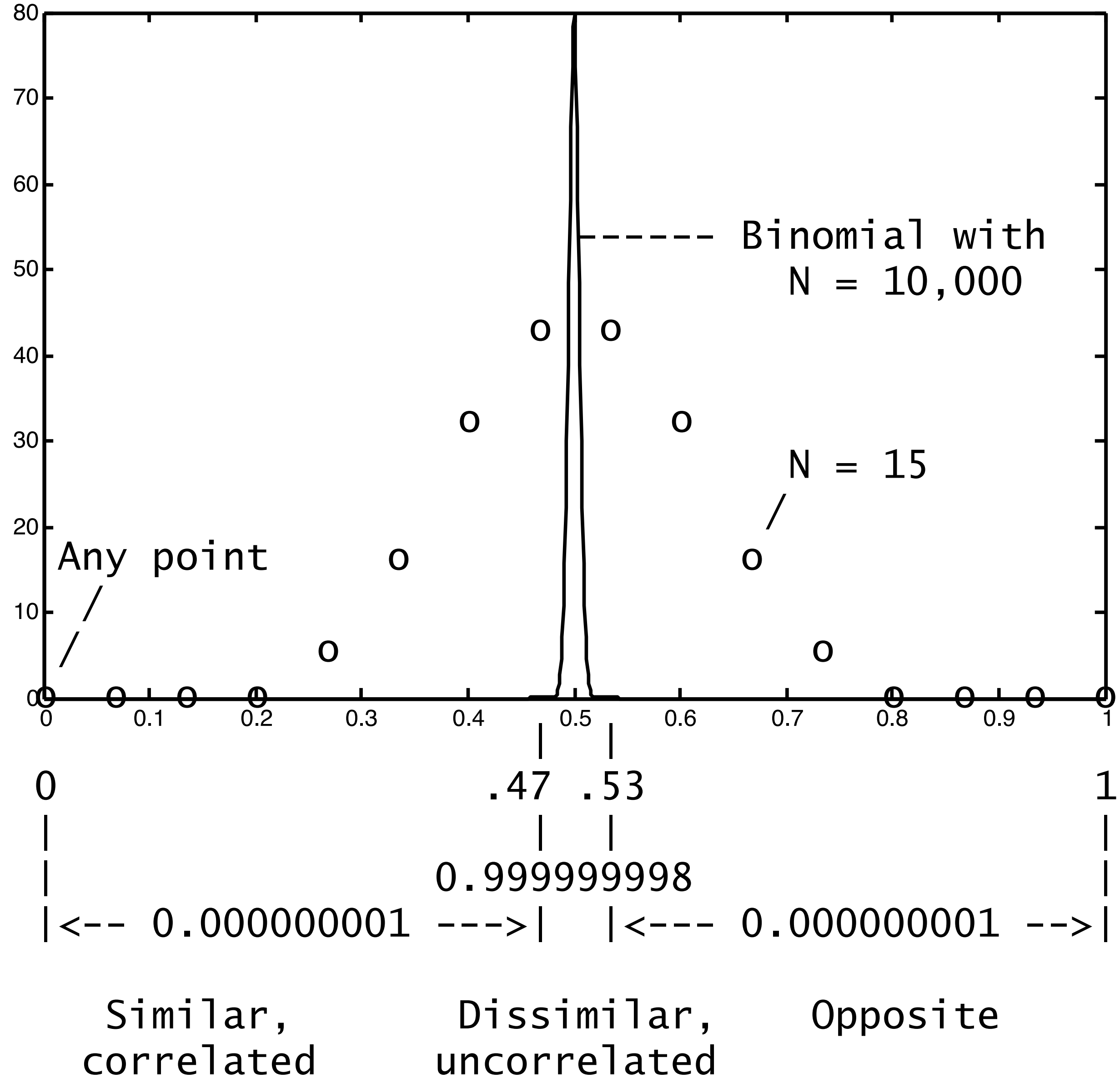


**FIGURE 4.** Binomial distribution

. Most vectors are dissimilar (*H* ~ 5,000, *h* ~ 0.5)

. A tiny fraction is closer than, say, 4,500 bits
-- *h* = 0.45 is 10 STDs from the mean
-- hence "very similar"

. Between pairs of dissimilar vectors (*h* ~ 0.5)
there are many that are *very similar to both*:

man ≈ fisherman ≈ fish ≈ lake

. These are properties of high-D vectors at large
-- binary, integer, real, complex vectors

. Called *Concentration of Measure*

```
          Hamming Distance/10,000
0.5     0.4     0.3     0,2     0,1      0
 |.......|.......|.......|.......|.......|
 |                             identical =
 |       |==== essentially the same =====|
 |   |===| very similar                  |
 | |=| correlated                        |
 |=| dissimilar                          |
 = orthogonal                            |
 |...|...|...|...|...|...|...|...|...|...|
 0  .1  .2  .3  .4  .5  .6  .7  .8  .9   1
            Pearson Correlation
```

**FIGURE 5.** "Similarity" of 10K-bit vectors based on random chance.

## Concepts as High-D Vectors

Old school:

- Concepts are represented by *disjoint features*

- Each feature is its own dimension of a high-D vector
  - e.g., age, sex, state, zip code, can swim, eligible to vote, speaks Chinese, married, number of children, ...
  - Grandmother cells

- The features constitute an **ontology**
  - i.e., concepts and categories of a subject area and relations among them
  - the list can grow indefinitely (>> 10K)

- There is no universal ontology
  - *any ontology will eventually box us in*

## Quasi-Orthogonality of High-D Vectors

In 10,000 dimensions there are 10,000 mutually orthogonal vectors but **billions of nearly orthogonal vectors**, i.e., dissimilar

. A randomly chosen vector is nearly orthogonal to any of a billion chosen so far

-- the number grows exponentially with dimensionality

. Each can represent an independent feature or concept

*Randomness is a major asset*

**Holographic Representation (Superposition)**

Overcomes the 10K limit on the number of features representable in 10K Bits

A single vector can represent

. a feature

. set of features

. structured composition of features

. concept

. . . .

## **Computing in Superposition**, an example

- Encode {$x$ = $a$, $y$ = $b$, $z$ = $c$} into a single superposition vector, *super-vector* S

- Retrieve the vector for $x$ from S

```
 X   = 10010...01    X and A are bound with XOR
 A   = 00111...11
----------------
X*A = 10101...10 -> 1 0 1 0 1 ... 1 0   (x = a)

 Y   = 10001...10
 B   = 11111...00
----------------
Y*B = 01110...10 -> 0 1 1 1 0 ... 1 0   (y = b)

 Z   = 01101...01
 C   = 10001...01
----------------
Z*C = 11100...00 -> 1 1 1 0 0 ... 0 0   (z = c)
                   ----------------------------
            Sum =  2 2 3 1 1 ... 2 0
       Majority =  1 1 1 0 0 ... 1 0  = S
```

**FIGURE 6a**, Encoding S = {*x* = *a*, *y* = *b*, *z* = *c*}

```
 "Un"bind S   S  =  1 1 1 0 0 ... 1 0
  with the    X  =  1 0 0 1 0 ... 0 1
  inverse    ------------------------
   of X      X*S =  0 1 1 1 0 ... 1 1  = A' ≈ A
                              |
                              v
                   .-----------------------.
                   |                       |
                   |  ASSOCIATIVE MEMORY   |
                   |        finds          |
                   |   nearest neighbor    |
                   | among stored vectors  |
                   |                       |
                   '-----------------------'
                              |
                              v
                    0 0 1 1 1 ... 1 1  = A
```

**FIGURE 6b.** Decoding:  What's the value of *x* in S?

**Summary of the Algorithm**

1. The variables and the values are represented by random 10K-bit **seed vectors** X, Y, Z, A, B, C

2. Variables are **bound** to their vales **with XOR** and the bound pairs are **combined with "addition"** (i.e., thresholded sum, majority)

3. The vector for *x* is retrieved with **XOR** and "**clean-up**"

## System of Computing with Super-Vectors

Ingredients

1. **Random-vector generator**:
   -- seed vectors

2. **Three operations** on vectors:
   -- Multiply, Add, Permute (MAP)

3. **Measure of similarity**:
   -- distance, cosine, Pearson correlation

These operations make it possible to do both
 . *rule-based symbolic processing* (GOFAI) and
 . *statistical learning from data*

**HOWEVER ...**

there is a limit to the amount of information
that can be stored in a single super-vector.
The limit is overcome by

4. **High-capacity memory for super-vectors**

   -- akin to memory for numbers and pointers
      in today's computers

**Functions of the Memory**

. Store and generate **sequences**: predict
-- *hetero-associative*

. **Identify** "noisy" vectors: clean-up
-- *auto-associative*

Modeled by neural-net associative memories

. Hopfield net
-- limited capacity

. Sparse Distributed Memory (SDM)
-- "unlimited" but inefficient

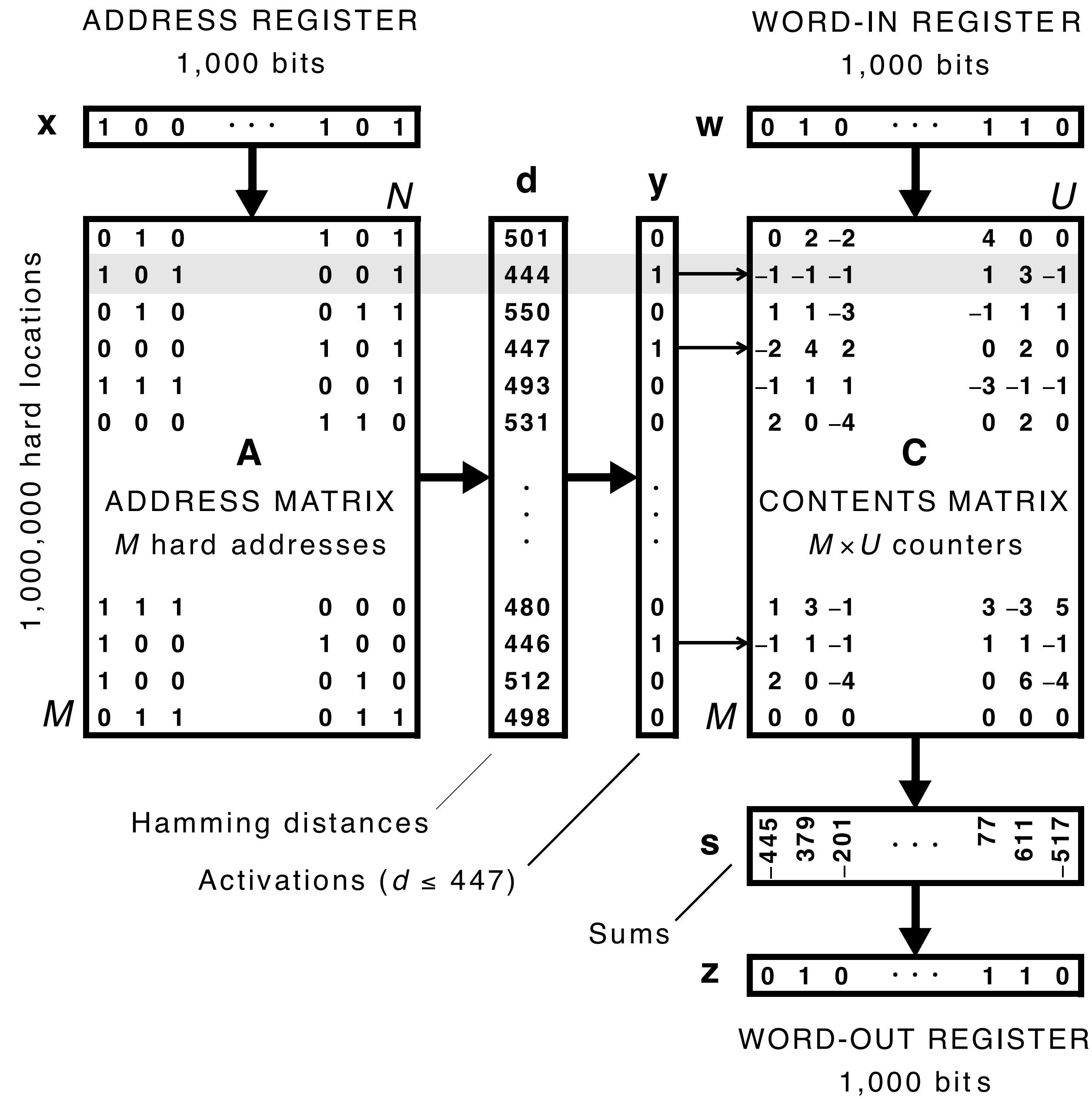


**FIGURE 7.** Spare Distributed Memory (SDM).

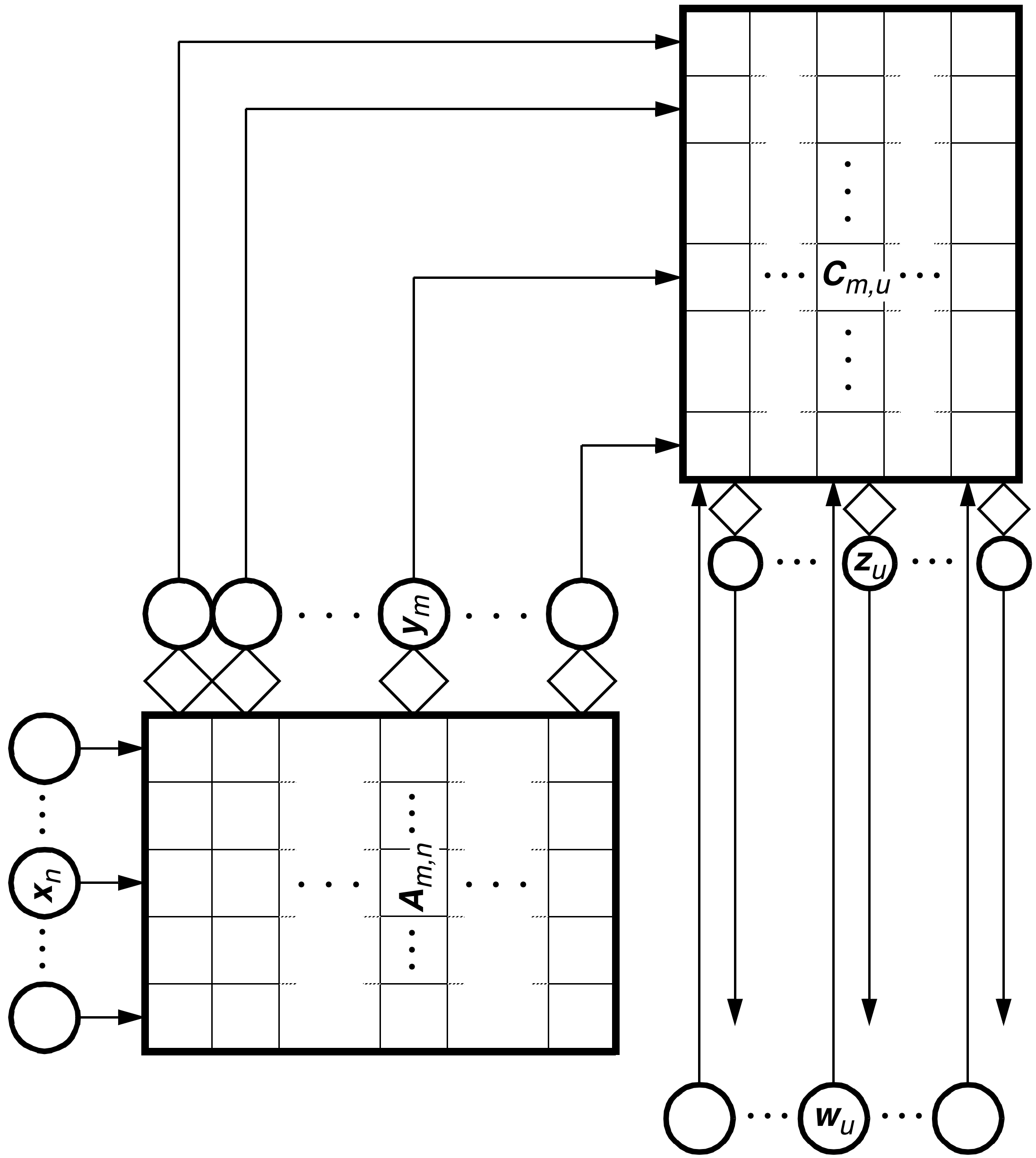


**FIGURE 8.** SDM morphs into cerebellum.

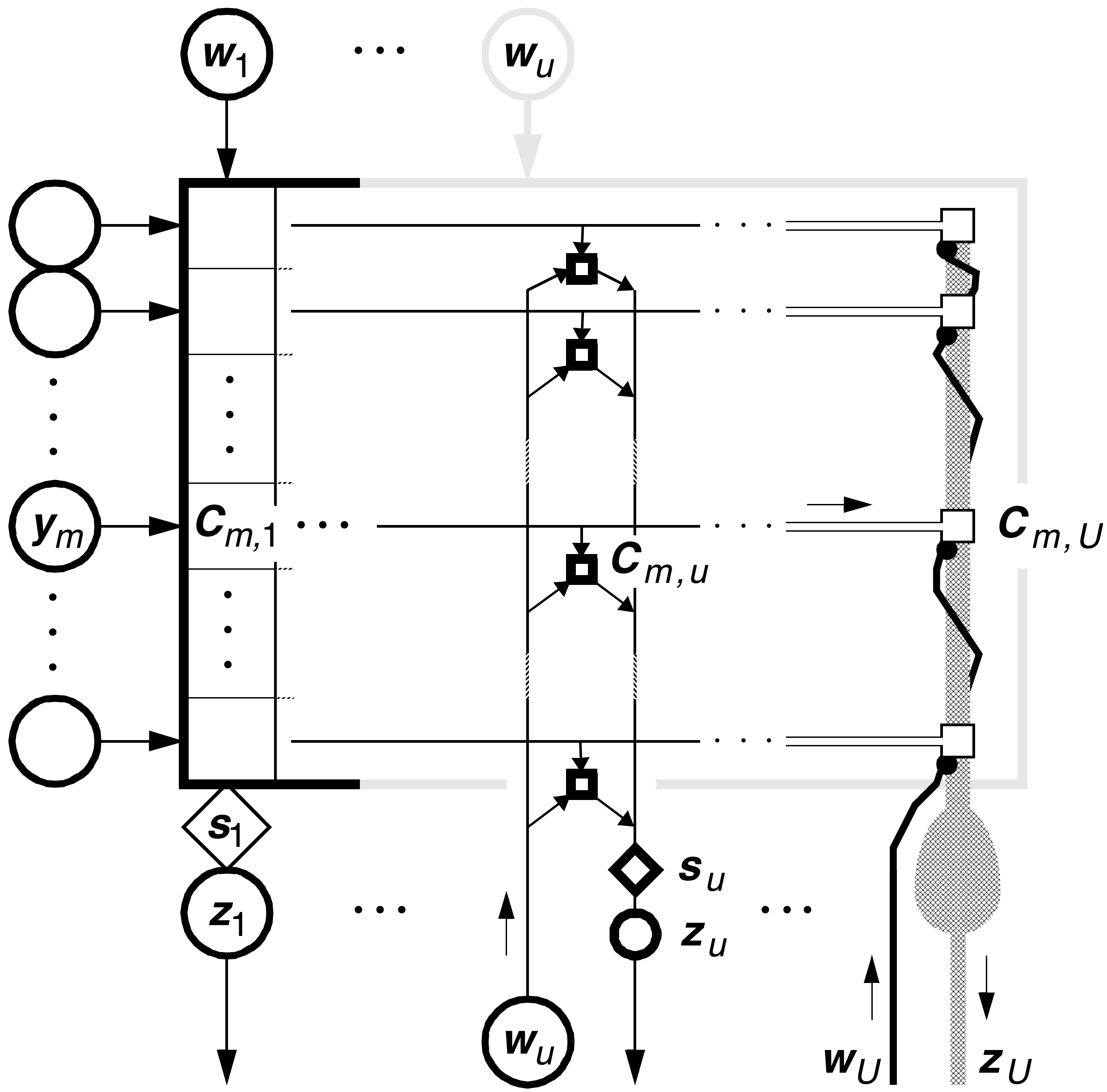


**FIGURE 9.** Weights, counters, synapses.

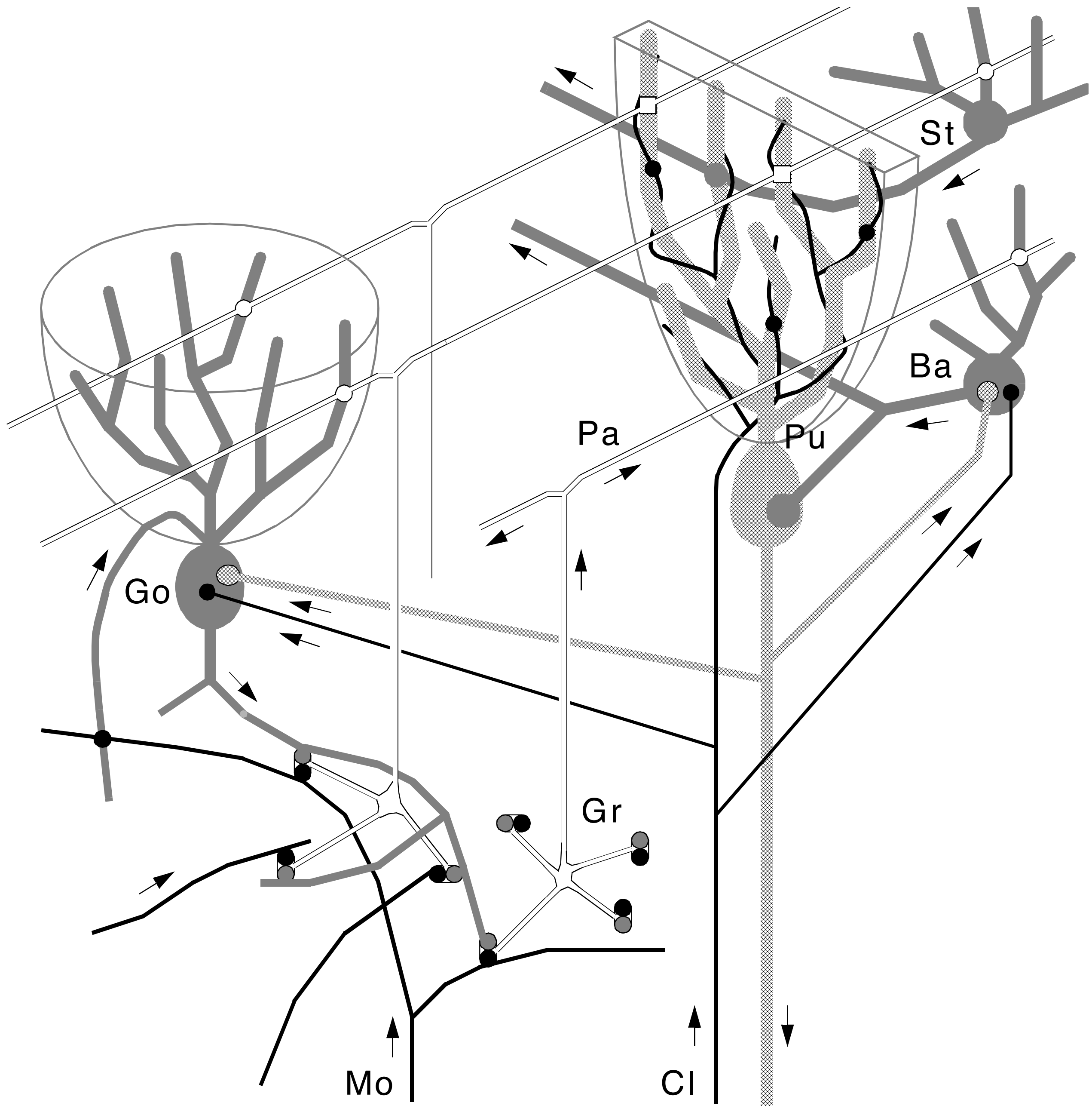


**FIGURE 10**. The cerebellum is more efficient than SDM.

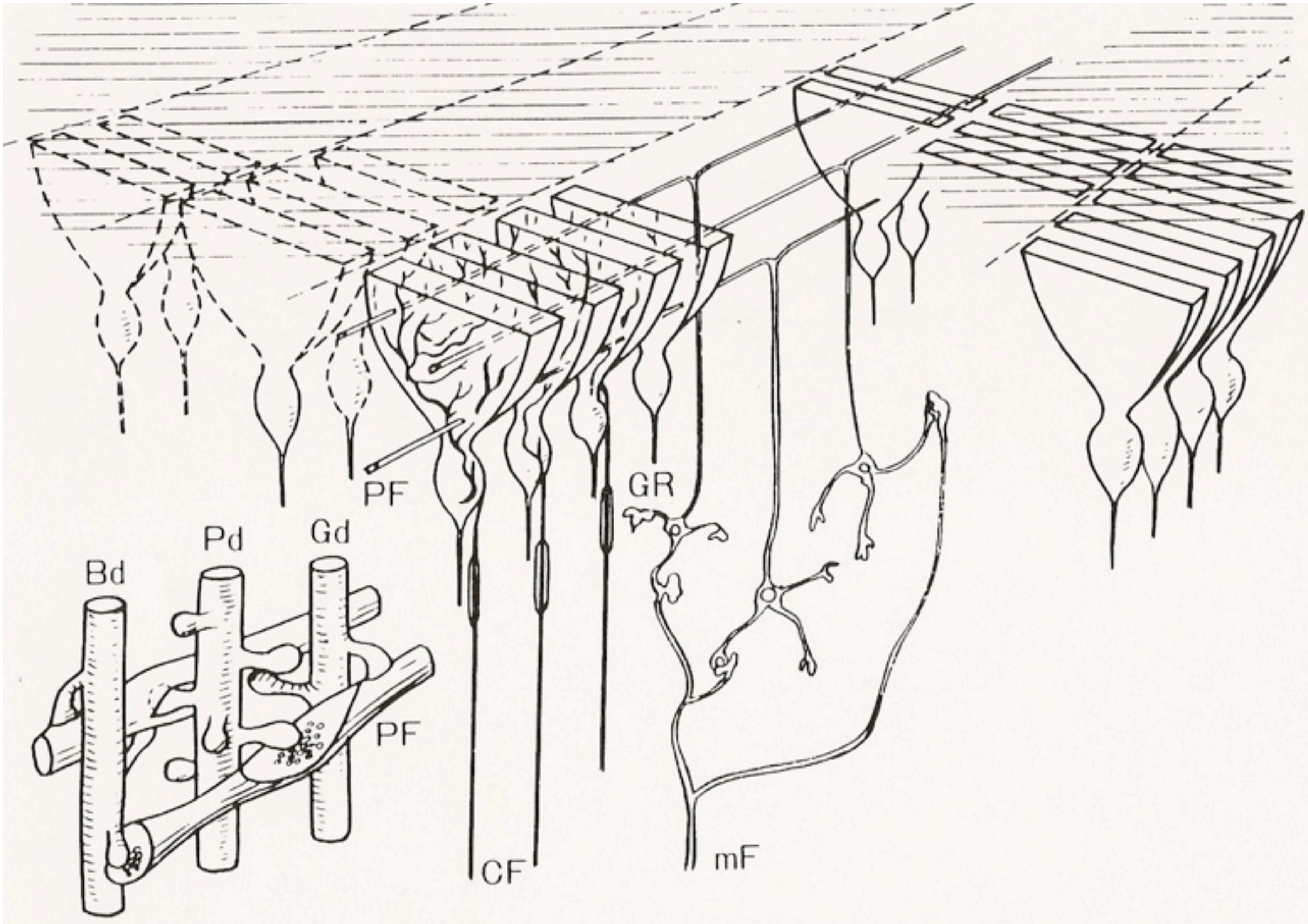


**FIGURE 3/11.** 3D organization of the "main" circuit.

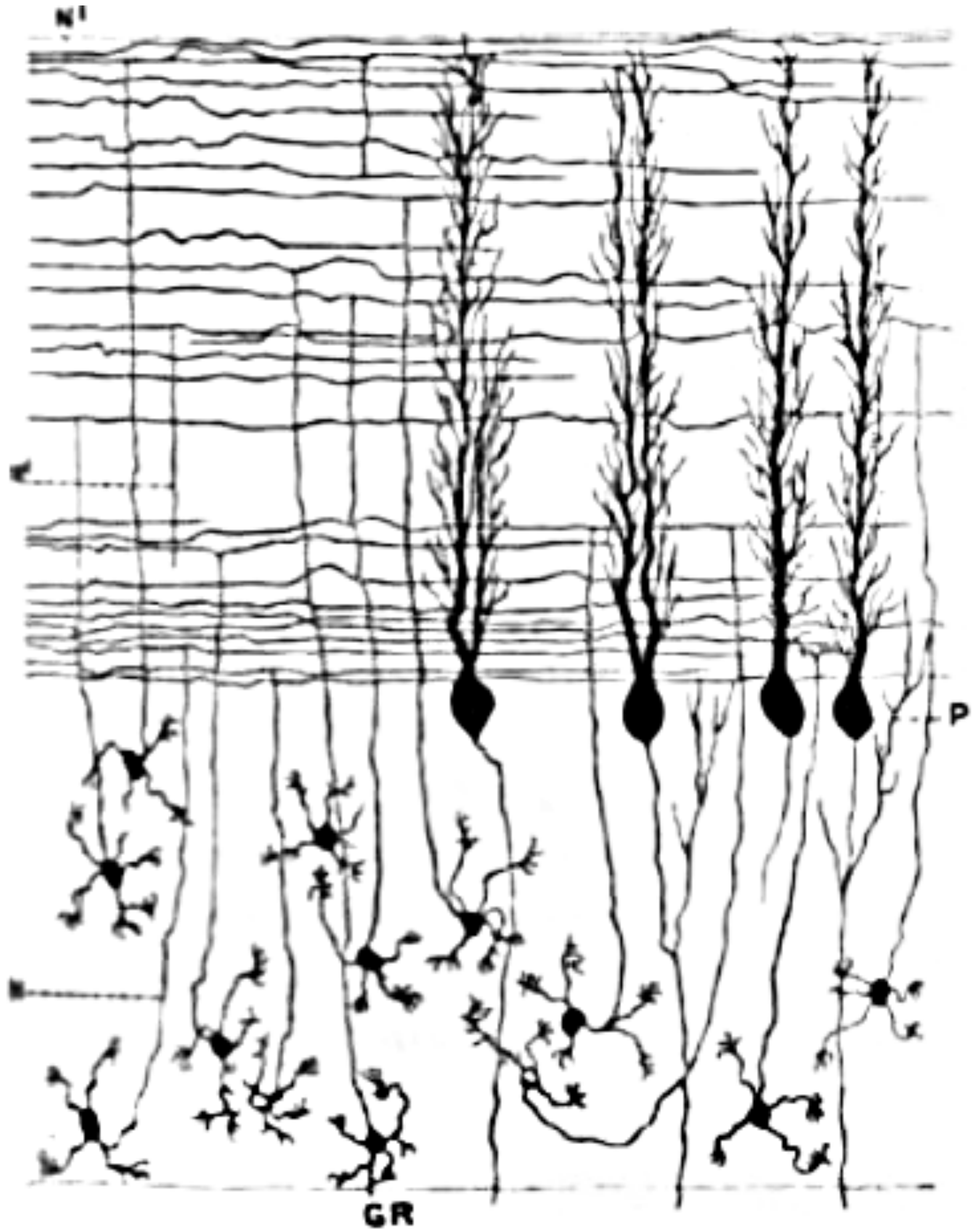


**FIGURE 12.** Side view: "Select lines" and "bit planes."

**Pseudo-Cerebellum**

. Building an associative memory for super-vectors is a major engineering challenge

. Nature appears to have solved it

. We can use the cerebellum as a source of ideas and guidance

*Calls for in-depth study of cerebellar circuits from engineering point of view* (Loebner, 1989)

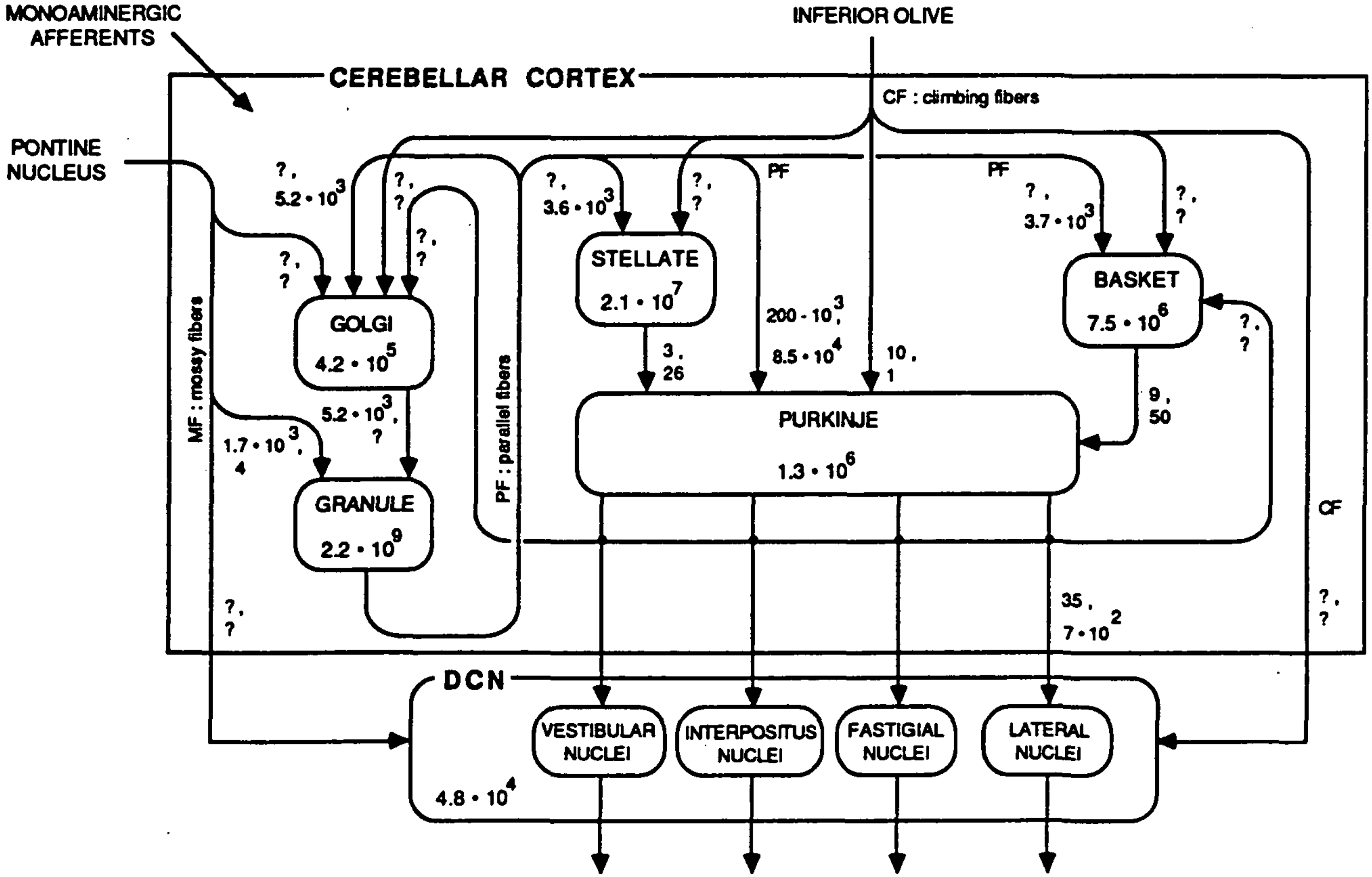


**FIGURE 13.** Cerebellar interconnect diagram.

**Is this Neuroscience?**

*Are differential equations physics?*

. No, but they are
. Mathematics to help us understand forces
of nature, i.e., *physics*

... and so, ... Is this neuroscience?

. No, but it is
. Computer Science to help us understand human
and animal behavior and traits, i.e., *brains*

## Concluding Remarks

*The nervous system and the brain are too complex to be understood without an adequate theory that serves as a framework in which to interpret our observations*

*Neuroanatomy and physiology are too important to be ignored in our theorizing about the brain and the mind*

**Origins**

. **Santiago Ramon y Cajal** (ca. 1900): anatomy

. **David Marr** (1969) and **James Albus** (1972): cerebellum as neural associative memory

. **PK** (1984): Sparse Distributed Memory

. **Tony Plate** (1994): computing in superposition (HRR = Holographic Reduced Representation)

. **Egon Loebner** (1989): interconnect diagram

. **Ross Gayler** (1998): significance of permutation (MAP = Multiply-Add-Permute)

. **Paxon Frady** (2017): computing with timing of spikes (complex-vector HRR)

**Grateful Acknowledgment of Support**

- **Intel** Strategic Research Alliance program on Neuromorphic Architectures for Mainstream Computing

- **NSF** 16-526: Energy-Efficient Computing: from Devices to Architectures (E2CDA). A Joint Initiative between NSF and SRC

- **AFOSR** FA9550-19-1-0241: Cognitive and Computational Neuroscience program, "Computing in Holographic Representation"

Simons Institute -- The Brain and Computation Reunion Workshop

https://simons.berkeley.edu/workshops/schedule/10785

2:00-3:00 PM on Wednesday June 17, 2019

**The Cerebellum as Neural Associative Memory**
Pentti Kanerva, UC Berkeley

**Abstract**

The cerebellum contains over half the neurons in the brain (the granule cells), as well as neurons with the largest number of modifiable synapses (the Purkinje cells). More than a century ago Santiago Ramon y Cajal mapped its circuits and left us with the puzzle of interpreting its function and operation. 70 years later David Marr (1969) and James Albus (1972) interpreted it as a neural associative memory. I will discuss this interpretation and its fit into a theory of computing with high-dimensional vectors. It turns out that computing with vectors resembles computing with numbers. Both need a large memory, to provide ready access to a lifetime's worth of information. I will also discuss the need to understand the cerebellum's connections to the rest of the nervous system in light of the theory of computing with vectors.