15-883: Computational Models of Neural Systems

Lecture 1.1:

Brains and Computation

David S. Touretzky
Computer Science Department
Carnegie Mellon University

Metaphors for the Nervous System

- Hydraulic network (Descartes): nerves = hoses
- Clockwork: systematic and representational
- Telephone switchboard: communication
- Digital computer (“electronic brain”): computational

Metaphors can serve as informal theories.

Helps to frame the discussion.

But lacks predictive power.

Meat Computers

The essential claim is this:

Brains perform computation.

Therefore:

Computational theories can give insight into brain function.

But what is “computation”?

When does it occur?

What is Computation?

There is no formal definition for what qualifies as “computation”. Lots of things do.

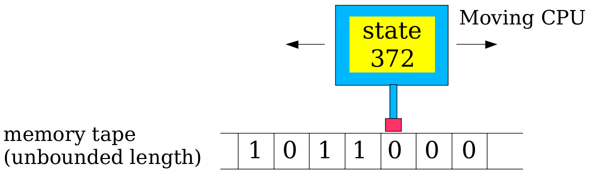
Abstract computational devices:

Turing machines, cellular automata, RAMs, lambda calculus, partial recursive functions, etc.

The Church-Turing hypothesis: any computable function can be computed by a Turing machine.

(If true, all computing formalisms are equivalent.)

Turing Machines



Finite number of states. Finite number of rules.

All rules are of form:
If state = nnn and current symbol = x, then:
Set current symbol to y,
Move left/right one square, and
Enter state mmm.

What's a Function?

- Abstract concept: functions map inputs to outputs
  - "domain": the set of possible inputs
  - "range": the set of possible outputs
- The input and output spaces can be vector spaces:

  [2, -7.4, 0.003] $\in$ [1.5, 2.8]

- Example: the human retina as 120 million receptors
- The optic nerve has 1 million axons.
- Retinal circuitry maps 120 million dimensional input space into a 1 million dimensional output space.

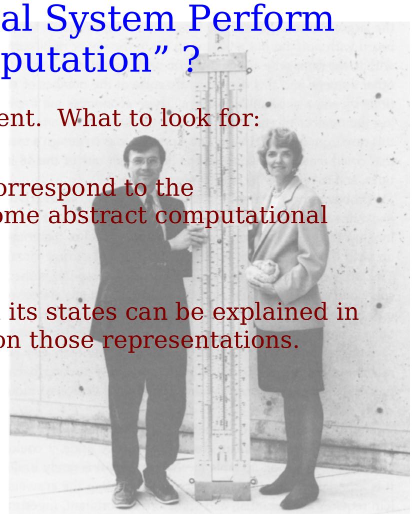
What Is A Computable Function?

- A mapping that can be described in terms of some set of rules, so that it has a finite description.
- The mapping itself can be infinite (e.g., defined over all the integers), as long as:
  - the description of the mapping rules is finite
  - the result can be determined in a finite number of steps
- Example of an uncomputable function: a truly random mapping of an infinite set, such as the integers — no finite description.

Physical Computation

- Turing machines, RAMs, etc., are theoretical abstractions.
- Brains are physical systems.
- So are digital computers.
- So are toasters.

Can A Physical System Perform "Computation" ?



It's a subjective judgment. What to look for:

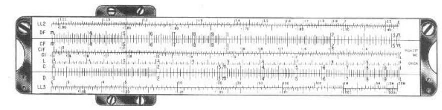
1) Its physical states correspond to the representations of some abstract computational system.

2) Transitions between its states can be explained in terms of operations on those representations.

9

Physical Computation: The Slide Rule

- Abstract function being computed: multiplication
  - Input: a pair of numbers (2D vector)
  - Output: a number (scalar)

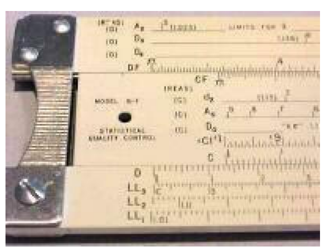


- Physical Realization:
  - First input = point on surface of the (fixed) D scale
  - Second input = point on surface of the (sliding) C scale
  - Output = point on surface of the (fixed) D scale

10

Slide Rule Computation Process

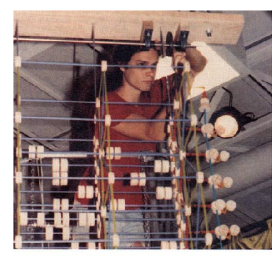
- Move the sliding C scale so that the digit "1" is above the first input on the D scale.
- Move the cursor so that the index is over the second input, on the C scale.
- Read the result under the index on the D scale.

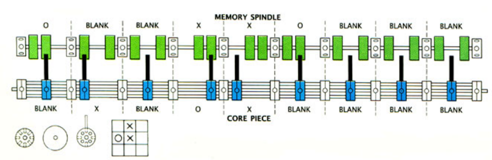


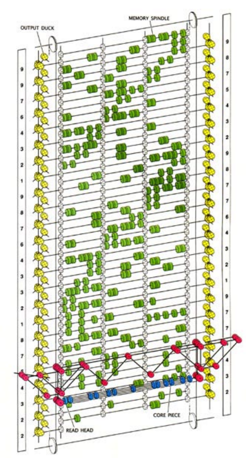
11

Tinkerytoy Tic-Tac-Toe Computer

Designed by Danny Hillis at MIT







12

When Does Computation Happen?

- If a slide rule falls in the forest, and nobody sees it, has computation taken place?
- YES: if the mapping is defined for us, then from our point of view, physical action = computation. No observer is required.
- NO: “Computation” is not a natural kind, like “plant”. It's a subjective judgment, like “weed” (a worthless plant). Worthless to whom?
- For something to be seen as computation, “interestingness” matters.

Systematicity Is Interesting

Let S be the set consisting of the following 24 items:

A1P	A1Q	A1R	A1S
B1P	B1Q	B1R	B1S
A2P	A2Q	A2R	A2S
B2P	B2Q	B2R	B2S
A3P	A3Q	A3R	A3S
B3P	B3Q	B3R	B3S

Let **f: S** $\in$ [0,1] be a function mapping S to the unit interval.

Can't say much about the domain: it's just a set of discrete points (“atoms”, “symbols”, etc.)

Structured Spaces

Let $\mathbf{S}_1$ = {A, B} $\mathbf{S}_2$ = {1, 2, 3} $\mathbf{S}_3$ = {P, Q, R, S}

Let **g:** $\mathbf{S}_1$ ™ $\mathbf{S}_2$ ™ $\mathbf{S}_3$ $\in$ [0,1] be a function from the Cartesian product space to the unit interval.

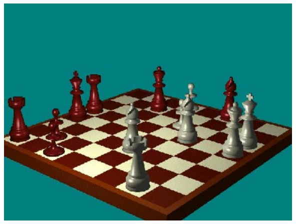
The domain of **g** has a concise description as a 3D space. The input points have internal structure which can be exploited in the definition of **g**.

<A,1,P> is more similar to <A,1,Q> than to <B,3,S>.

How to Play Chess...

Input space: board configuration (64D vector)

Output space: a legal move (one item from finite set)



Physical representation of input:

A chess board with plastic pieces

Input mapping to physical system:

Plastic pieces should be laid out on the physical board so as to agree with the input vector.

How to Play Chess... with an Anvil

Computational process:

Drop a 2 ton anvil on the chess board from a height of 5 feet.

Physical representation of the output:

Pattern of smooshed plastic shapes embedded in the board.

What's wrong with this computation?

17

Interpreting the Debris

Output mapping:

If board looks like any of:  ...
the result is P-K4.

If board looks like any of:  ...
the result is P-Q3.

What's wrong?

- The mapping requires trillions of smooshed board images.
- The anvil isn't exploiting any regularities of the domain.
- All the work (way too much) is being done by the output mapping.

18

Systematic Interpretability

What make for a good example of computation?

- Physical representations must be <u>systematically interpretable</u> as input and output values.

  The mapping must be <u>concise</u>.

- Arbitrary mappings don't provide a satisfying analogy between the abstract function and the physical system.

- Interesting functions are ones that cleverly exploit the regularities of the domain.

19

Do Toasters Compute?

They compute the results of applying sustained heat to rectangular chunks of carbohydrates.

(They make toast.)

But is the process “interesting” enough to be called computation?

What if there were some really interesting, systematic mapping? Then our view of toasters might change.

20

Do Brains Compute?

Most scholars believe the answer is “yes”.

Some consider this conclusion demeaning.

Some try to find reasons the answer could be “no”.

Example: if unpredictable quantum effects played a crucial role in what brains do, then the result would not be describable as a computable function.

Computational Processes Posited in the Brain

- Table lookup.
- Matrix memory (associative memory).
- Competitive learning.
- Self-organizing maps.
- Gradient descent error minimization learning.
- Temporal difference learning.
- Dynamical systems (attractor networks, parallel constraint satisfaction).

How Big Is This Computer? Some Numbers

	Neurons	Synapses
Humans	10^{12}	10^{15}
Rats	10^{10}	10^{13}
1 mm^3 of cortex	10^5	10^9

A cortical neuron averages 4.12 ™10^3 synapses (cat or monkey.)

Cherniak (1990) estimates 39,000 synapses/neuron in some parts of cortex.

In primary visual cortex, cells are dense (100,000 per mm^3), so fewer synpases per cell: 1.17 ™10^3

Demystifying the Brain

- The cerebellum, concerned with posture and movement (and...?), contains four times as many neurons as the cortex, seat of language and conscious reasoning.
- There are roughly 10^{13} synapses in cortex. Assume each stores one bit of information. That's 1.25 terabytes.
- The Library of Congress (80 million volumes, average 300 typed pages each) contains about 48 terabytes of data (Cherniak, 1990).
- The brain is complex, but not infinitely so.