BIO LOGIC: Biological Computation

Kay Kirkpatrick, Math 490

2019
Kay Kirkpatrick, Math 490

Urbana, Miami and Mascouten lands

Movie still from the Yonath lab; next slide from Cambridge U.
Turing 1948: Intelligent Machinery

3. Varieties of machinery

It will not be possible to discuss possible means of producing machinery without introducing a number of technical terms. Let us define:

- ‘Discrete’ and ‘Continuous’ machinery. We may call a machine discrete if it is natural to describe its possible states as a discrete set. We call it continuous if on the other hand form a continuous manifold, and its behaviour be regarded as continuous, but when it is possible to repeat it in a cycle usually best to do so. The states of discrete machinery will be discrete ‘configurations’.

- ‘Controlling’ and ‘Active’ machinery. Machinery may be described as ‘controlling’ if it only deals with information. In practice this is not much use as saying that the magnitude of the machine’s effects may be put as please, so long as we do not introduce confusion through these means. e.g. ‘Active’ machinery is intended to produce some definite physical effect.

Examples

- A Bulldozer is continuous active
- A Telephone is continuous controlling
- A Brunsviga* is discrete controlling
- A Brain is probably continuous controlling
- Similar to each discrete machine
- The ENIAC, ACE etc. are discrete controlling
- A Differential Analyser is continuous controlling

We shall mainly be concerned with discrete controlling machines. We have mentioned, brains very nearly fall into this class, and there seems very reason to believe that they could have been made to fall naturally into a discrete variety of configuration and change in their essential properties. However, the property of being fully discrete.

* Editor’s note. The Brunsviga was a popular desk calculating machine.

Courtesy Copeland, Complete Turing
‘Controlling’ and ‘Active’ machinery. Machinery may be described as ‘controlling’ if it only deals with information. In practice this condition is much the same as saying that the magnitude of the machine’s effects may be as small as we please, so long as we do not introduce confusion through Brownian movement etc. ‘Active’ machinery is intended to produce some definite physical effect.

**Examples**

- A Bulldozer is Continuous Active
- A Telephone is Continuous Controlling
- A Brunsviga is Discrete Controlling
- A Brain is probably Continuous Controlling, but is very similar to much discrete machinery
- The ENIAC, ACE etc. Discrete Controlling
- A Differential Analyser Continuous Controlling
The brain processes information AND is active:

- ions, neurotransmitters, neuromodulators, 
- hormones, remodeling synapses, movement, fields.
His 1950 Imitation Game relies on this mistake courtesy wikipedia

Turing Test involves just info input & info output.
His 1950 Imitation Game relies on this mistake
courtesy wikipedia

Turing Test involves just info input & info output.

Same with the Chinese Room Argument objection.
Need new: Active + Information Machines that...

Process info AND produce definite physical effects

Examples: Brains, organs, robots, ribosomes ...
Need new: Active + Information Machines that...

Process info AND produce definite physical effects

Examples: Brains, organs, robots, ribosomes ...

Computers? (require user)

Artificial neural networks? (some act)
A + I Machines may be better than Turing machines:

Brains, etc., compute and act simultaneously.

Robots have a communication bottleneck.
A + I Machines may be better than Turing machines:

Brains, etc., compute and act simultaneously.

Robots have a communication bottleneck.
Statistical mechanics might help us figure out the brain
Statistical mechanics might help us figure out the brain.

microscopic first principles $\leadsto$ zoom out $\leadsto$ Macroscopic states

Courtesy Greg L and Digital Vision/Getty Images
Outline

1. What is the microscopic unit of computation?
2. Unit properties: information & computation.
3. Consequences of this new approach.
What are microscopic computational units in the brain?

Neurons are mesoscopic & complicated.

McCulloch-Pitts and von Neumann claimed: neuron $\simeq$ vacuum tube (wrong)
What are microscopic computational units in the brain?

Neurons are mesoscopic & complicated.

McCulloch-Pitts and von Neumann claimed:
neuron $\simeq$ vacuum tube (wrong)

If they were right, the iPhone could model a brain.
S. Grant: a synapse is a computer > vacuum tube

So a neuron $\geq 10^4$ synapses $\gg$ vacuum tube

Instead?
S. Grant: a synapse is a computer $>\ $vacuum tube

So a neuron $\geq 10^4$ synapses $\gg$ vacuum tube

Instead? Proteins have similar complexity to vacuum tubes, and ribosomes help make proteins.
Ribosomes synthesize proteins from mRNA genes

Courtesy Jay Swan
How to model a ribosome mathematically? (Take 1)

Detailed: Molecular Dynamics. But movie:

Reductive: TASEP. No information. But theory.
How to model a ribosome, Take 2

Caetano-Anollés & Caetano-Anollés 2015:
As a Universal Turing Machine in the cell.

As a finite-state automaton.
How to model a ribosome, Take 2


As a finite-state automaton.

Reductive and info/computational, not active.

AIM above pure info, but first info details...
The ribosome as an information channel

proto-code DNA $\xrightarrow{\text{transcription, regulation}}$ mRNA $X^*$

channel: Ribosome

folded protein $\tilde{Y} \xleftarrow{\text{post-translational modifications}}$ polypeptide $Y^*$
The ribosome operates under its Shannon capacity

**Theorem (Inafuku-K.-Osuagwu 2019):**
Ribosome’s Shannon capacity is $\geq 266$ bits per second. Cf: observed rates $\sim 24$ to $120$ bits per second.

Proof uses the uniform distribution.

This is why ribosomes are $\sim 99.99\%$ accurate.
Enzymes operate under their Shannon capacities

**Theorems (D. Inafuku 2019):** Theoretical capacities & observed rates (units are bits/sec):

<table>
<thead>
<tr>
<th>Enzyme name</th>
<th>Cap.</th>
<th>Actual rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA polymerase</td>
<td>1554.2</td>
<td>1498</td>
</tr>
<tr>
<td>RNA II polymerase</td>
<td>191.9</td>
<td>160</td>
</tr>
<tr>
<td>KcsA ion channel</td>
<td>$3 \times 10^7$</td>
<td>$2 \times 10^7$</td>
</tr>
</tbody>
</table>
Input mRNA tape $X^*$ is string of codons in $\Sigma := \{A, C, G, U\}^3$

$$\Sigma^* := \{ \text{words with letters in } \Sigma, \text{ incl. empty word } \epsilon \}.$$
Input mRNA tape $X^*$ is string of codons in $\Sigma := \{A, C, G, U\}^3$

$$\Sigma^* := \{\text{words with letters in } \Sigma, \text{ incl. empty word } \epsilon\}.$$ 

Output polypeptide $Y^*$ is a string of **physical blocks** in

$$\Delta^* := \{\text{strings of blocks in } \Delta := \{b_1, b_2, \ldots, b_{20}\}\}.$$
Notation to get beyond pure computation

Input mRNA tape $X^*$ is string of codons in $\Sigma := \{A, C, G, U\}^3$

$\Sigma^* := \{\text{words with letters in } \Sigma, \text{ incl. empty word } \epsilon\}.$

Output polypeptide $Y^*$ is a string of physical blocks in $\Delta^* := \{\text{strings of blocks in } \Delta := \{b_1, b_2, \ldots, b_{20}\}\}.$

$Y^*$ contains info and is also a physico-chemical structure.
Turing defined a 2-tape automatic a-machine

Consists of six parts: \( M := \langle Q, \Sigma, \Delta, \delta, q_0, F \rangle \)

- State space \( Q \)
- \( \Sigma \) input alphabet, and input program \( X^* \in \Sigma^* \)
- \( \Delta \) output symbols, and output tape \( Y^* \)
Turing defined a 2-tape automatic a-machine

Consists of six parts: \( M := \langle Q, \Sigma, \Delta, \delta, q_0, F \rangle \)

- State space \( Q \)
- \( \Sigma \) input alphabet, and input program \( X^* \in \Sigma^* \)
- \( \Delta \) output symbols, and output tape \( Y^* \)
- Initial state \( q_0 \)
- Final/accepting states \( F \subset Q \)
- Transition function (rules about updating state/tape) \( \delta : (Q \setminus F) \times \Sigma \rightarrow Q \times \Sigma \times \Delta \times \{L, R\} \)
We define an automatic biochemical abc-machine

An **abc-machine** \( M_{T,A} = \langle Q, \Sigma, \Delta, \delta, q_0, F, T, A \rangle \) has:

- **Q, \Sigma, q_0, F** as before; input program \( X^* \in \Sigma^* \)
- **State space** \( Q \) with initial state \( q_0 \), final states \( F \subset Q \)
- **Transitions** \( \delta : (Q \setminus F) \times \Sigma \rightarrow Q \times \Sigma \times \Delta \times \{L, R\} \)
We define an automatic biochemical abc-machine

An **abc-machine** $M_{T,A} = \langle Q, \Sigma, \Delta, \delta, q_0, F, T, A \rangle$ has:

- $Q, \Sigma, q_0, F$ as before; input program $X^* \in \Sigma^*$
- State space $Q$, with initial state $q_0$, final states $F \subset Q$
- Transitions $\delta : (Q \setminus F) \times \Sigma \to Q \times \Sigma \times \Delta \times \{L, R\}$

- $\Delta$ output **blocks**, and $Y^*$ is **block-string** output
- $T$ a random stopping time, when program $X^*$ **halts**
- $A$ an auxiliary machine, optimizing a bio-chemical energy functional, e.g., protein-folding
Diagram for the ribosome as an abc-machine $M_{T,A}$

If the RNA degrades at time $T(X^*, \omega)$, transition to HALT.

Auxiliary machine $A$ folds the polypeptide: $Y^* \rightsquigarrow \tilde{Y}$
Observation (K.-Osuagwu 2018): $M_{T,A}$ is equivalent to a 2-tape TM with 2 oracles: protein-folding machine $A$ for output $\tilde{Y}$, and stopping time $\{t < T\}$.

The ribosome $M_{T,A}$ is an A+I machine.
The ribosome is more than just a Turing Machine (TM)

**Observation (K.-Osuagwu 2018):** $M_{T,A}$ is equivalent to a 2-tape TM with 2 oracles: protein-folding machine $A$ for output $\tilde{Y}$, and stopping time $\{t < T\}$.

The ribosome $M_{T,A}$ is an A+I machine.

Protein product $\tilde{Y}$ is an A+I machine too.

(Modified input pieces can act too: e.g., ATP, GTP.)
Better than TMs on Decidability/Computability

Abc-machines avoid the Halting Problem by stopping at $T$. 
Better than TMs on Decidability/Computability

Abc-machines avoid the Halting Problem by stopping at $T$.

A non-deterministic Turing Machine (NDTM) can approx. but not simulate $T$, which is truly random & non-computable.
Abc-machines are possibly higher complexity than TMs

If protein folder $A$ is NP-complete, then it is in class $\textbf{NP}$.

Self-avoiding random walk model of $A$ is NP-complete (Berger and Leighton 1998)
Consequences for the brain

Individual computational units are more than TMs.
Consequences for the brain

Individual computational units are more than TMs.

Conjecture: Brain is more than a TM. Embodiment.
When can we model the whole brain? (Weak AI)

The brain has $\sim 10^{11}$ cells (neurons and glia).

Each cell has $\sim 10^6$ ribosomes and $\sim 10^7$ proteins.
When can we model the whole brain? (Weak AI)

The brain has $\sim 10^{11}$ cells (neurons and glia).

Each cell has $\sim 10^6$ ribosomes and $\sim 10^7$ proteins.

Total: $\sim 10^{18}$ units, ea. $\sim 100$ binary ops per sec.
When can we model the whole brain? (Weak AI)

The brain has $\sim 10^{11}$ cells (neurons and glia).

Each cell has $\sim 10^{6}$ ribosomes and $\sim 10^{7}$ proteins.

Total: $\sim 10^{18}$ units, ea. $\sim 100$ binary ops per sec.

Need zettascale ($10^{21}$ flops) or yottascale ($10^{24}$) computing, expected around 2030 and 2040.
When can we model the whole brain? (Weak AI)

The brain has \( \sim 10^{11} \) cells (neurons and glia).

Each cell has \( \sim 10^{6} \) ribosomes and \( \sim 10^{7} \) proteins.

Total: \( \sim 10^{18} \) units, ea. \( \sim 100 \) binary ops per sec.

Need zettascale (\( 10^{21} \) flops) or yottascale (\( 10^{24} \)) computing, expected around 2030 and 2040.

2-week weather modeling may require zettascale.
Why are artificial “neural networks” successful?

Maybe not because they’re like brains, but like something else...
Maybe artificial neural networks are successful...

Because they’re like biochemical networks in E. coli, which can basically do Stochastic Gradient Descent.
But artificial neural networks fail spectacularly

An adversarial example:

```
“panda”
57.7% confidence

+ ε

= “gibbon”
99.3% confidence
```

Courtesy Open AI
Key takeaway: “Neural network” is a misnomer

Nodes (neurones) are simpler than real neurons.

More takeaways

Biological computation is more sophisticated than we thought.
Biological computation is more sophisticated than we thought.

True AI may be harder than we thought.
Thanks

NSF CAREER Award DMS-1254791, Simons Sabbatical Fellowship

faculty.math.illinois.edu/~kkirkpat/
What about DNA computing?