5 Computation in Physical Systems

The universe has been estimated to have performed 10^{120} operations on 10^{90} bits [Lloyd, 2002]. To understand how it's possible to make such a statement, and to understand this book, it's necessary to understand the connection between computation and physics.

5.1 ANALOG

5.1.1 Linear

A *linear function* has the properties of *additivity* $f(x+y) = f(x)+f(y)$ and *homogeneity* $f(\alpha x) = \alpha f(x)$. For a scalar function of one variable, that's satisfied by $f(x) = ax$. Closely related is $f(x) = ax + b$, which is *affine* because of the constant. For a function of many variables, linearity becomes a matrix times a vector $f(\vec{x}) = \mathbf{A}\vec{x}$. Mathematically, these describe operations including filters and transforms [Gershenfeld, 1999a]; in Chapter 10 we'll see that they can also be performed by lenses.

A *linear program* (*LP*) seeks to maximize (or minimize) a linear function with constraints:

$\max \vec{c} \cdot \vec{x}$

$\mathbf{A}\vec{x} \geq \vec{b}$

$\vec{x} \geq 0$

These are ubiquituous in logistics and economics. They're a special case of the general category of *mathematical programs*, which minimize arbitrary functions with constraints.

A linear *system of equations*:

$\mathbf{A}\vec{x} = \vec{b}$

can be solved by matrix inversion [Gershenfeld, 1999a]:

$\vec{x} = \mathbf{A}^{-1}\vec{b}$

It can instead be converted to a *quadratic program* (*QP*) with squared terms by minimizing the square magnitude of the error:

$|\mathbf{A}\vec{x} - \vec{b}|^2$

QPs can be solved effeciently because they are *convex*, with a unique global optimum [Boyd & Vandenberghe, 2004]

Since physical systems can minimize energy it should not be surprising that they can solve mathematical programs, such as by analog electrical circuits with resistors representing the coefficients and diodes the constraints [Dennis, 1959, Vichik & Borrelli, 2014], and by networks of springs (Problem ?).

5.1.2 Nonlinear

A *nonlinear function* is one that is, well, not linear.

cant solve in general, runs afoul of computability (to come)

notable nonlinear function *McCulloch-Pitts* [McCulloch & Pitts, 1943] synapse model

binary inputs neuron fires (outputs 1) if weighted sum of inputs about a threshold, othwise outputs 0

$\sum_i w_i x_i \geq \theta$

Rosenblat Perceptron [Rosenblatt, 1958] learning rule

single layer perceptron linearly separable can't do XOR function truth table, gate Minsky [Minsky & Papert, 1969] two layers can map to separable

threshold needs slope for training *backpropagation* [Rumelhart *et al.*, 1986]

sigmoid ReLU

universal approximation theorem [Hornik *et al.*, 1989]

deep learning [Sejnowski, 2020]

breadth vs depth [Telgarsky, 2016, Poggio *et al.*, 2017]

GPU tensor cores MAC multiply-accumulate

memristors neuromorphic Chapter 12

diverged from, converged with brain [Schyns *et al.*, 2022]

5.1.3 Differential

differential equations

most common physics representation

this book full of them

differential analyzer [Bush, 1931, Shannon, 1941]

integration by disk rotation gear and shafts to add, subtract, multiply

analog computers Chapter 6 integrate, differentiate simple, interactive long before digital real-time

5.2 DIGITAL

analog errors accumulate

notable student Shannon

5.2.1 Logic

Boolean logic

best Masters thesis ever [Shannon, 1937]

introduced logic operations from relays, logic minimization

truth tables, symbols

NOT $\neg a = \overline{a}$

AND $a \wedge b$

OR $a \vee b$

XOR $a \oplus b$

NAND $\overline{a \wedge b}$

GF(2) finite field

XOR addition x1+x2 mod 2 linear

AND multiplication x1*x2 nonlinear

not universal need negation NAND 1-x1*x2 universal Problem ?

minimization of Boolean functions [McCluskey, 1956] *Quine-McCluskey* philosopher Willard Van Orman Quine self-replicating computer programs

implemented in sums of products PLD, FPGA

reversible Logic

Fredkin gate [Fredkin & Toffoli, 1982] controlled SWAP

Toffoli gate [Toffoli, 1980] controlled-controlled-NOT generalized XOR for universality

theoretical, returned in low power, essential in quantum

5.2.2 Computation

combinatorial vs sequential

Turing machine [Turing, 1936]

tape with cells containing symbols, head read and write symbols, move the tape left or right, state register, table of instructions for state and symbols

arguably the first designed: Babbage Analytical Engine 1838 [Bromley, 2008]

arguably the first realized: Konrad Zuse Z3 1941 [Zuse, 1993]

Church-Turing thesis Turing machine universal [Kleene, 1967]

universality ubiquitous logic, memory, connectivity

fluid droplets [Prakash & Gershenfeld, 2007]

billiard balls [Margolus, 1984]

modeled with *cellular automata CAs* [Ulam, 1962]

viewed as fundamental, finite information in finite space [Fredkin, 1990]

can model CAs with differential equations [Omohundro, 1984]

5.2.3 Complexity

in practice feasible N vs $N \log N$ vs N^2 [Gershenfeld, 1999a] 1 ns

10^3 steps 10^{-6} sec 10^{-5} sec 10^{-3} sec

10^9 steps 1 sec 30 sec 10^4 days

more fundamental, is it even possible?

Turing halting problem

assume H solves halting problem

H(P,I) = 1 if program halts on input, 0 otherwise

D(P) = 1-H(P,P), does opposite

H(D,D) = 1 means D(D) halts, therefor D doesn't halt

H(D,D) = 0 means D(D) does not halt, therefore D halts

contradiction!

diagonalization argument

undecidability

undecidable in dynamics [Moore, 1990]

in between practical and impossible is feasible
polynomial vs exponential
P solve polynomial time
NP nondeterministic, verify polynomial time requires decision
NP-complete every problem reduced in poly time
SAT Boolean satisfiability graph coloring not touching Hamiltonian path visits each vertex once
profound result Cook's Thm [Cook, 1971] SAT is NP-complete
NP-hard at least as hard, not in NP-complete
halting TSP minimization vs verification spin glass minimization local vs global minima cooling, annealing phase transitions in complexity [Kirkpatrick & Selman, 1994]
complexity taxonomy [Papadimitriou, 2003] over 500 classes [Aaronson, 2002]
many distinctions unphysical e.g. NP, echo in quantum physical complexity class causality, finite [Gershenfeld, 2011] propagation, interaction [Gershenfeld, 2015]

5.2.4 Reliability

error correction threshold theorem
majority voting example
error ϵ
majority vote 3 measurements $3\epsilon^2$ error
vote on vote $3 \times 3 = 3^2$ measurements $3(3\epsilon^2)^2 = (3\epsilon)^4/3$ error
recurse
3^N measurements $(3\epsilon)^{2^N}/3$ error
Reliable circuits using less reliable relays [Moore & Shannon, 1956]
Probabilistic logics and the synthesis of reliable organisms from unreliable components [von Neumann, 1956]
architecture
A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology [Heath *et al.*, 1998]
correct at end analog logic [Vigoda *et al.*, 2006]
Landauer erasure dissipation previous chapter
Bennett reversible fluctuation-dissipation [Bennett, 1982]

5.3 FABRICATION

brain vs supercomputer [Gershenfeld, 2020]
Supercomputer exaflop 1e18 ops/s
Brain 1e15 synapses, 100Hz firing, 1e17 ops/s
exaflop 1e7 cores, 1e8 transistors/core = 1e15 transistors petabyte 1e15 bytes, 10 transistors/byte = 1e16 transistors
fab 1e4 wafers/month * 1e5 mm2 wafer * 1e8 transistors/mm2 = 1e10 parts/s
person 1e13 cells * 1e5 ribosomes/cell * 1 Hz = 1e18 parts/s = 100 supercomputers/s
can compute with molecules

DNA molecule provides a computing machine with both data and fuel [Benenson *et al.*, 2003],

bio building blocks

bio error correction

morphogenesis

The Chemical Basis of Morphogenesis [Turing, 1952]

Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo [Wagner *et al.*, 2018]

Theory of Self-Reproducing Automata [von Neumann, 1966]

Life in Life [Bradbury, 2012]

5.4 SELECTED REFERENCES

[Moore & Mertens, 2011] Moore, Cristopher, & Mertens, Stephan. (2011). *The nature of computation.* Oxford University Press.

A very readable tour through the theory of computation.

[Gershenfeld *et al.*, 2017] Gershenfeld, Neil, Gershenfeld, Alan, & Cutcher-Gershenfeld, Joel. (2017). *Designing reality: How to survive and thrive in the third digital revolution.* Hachette UK.

The implementation and implications of the connection between computation and fabrication.

5.5 PROBLEMS

solve network of springs as mathematical program

derive all logic gates from a NAND gate implement a NAND gate in reversible logic implement a serial adder in reversible logic

spatial program