1

COGS1460
Computational Cognitive Neuroscience

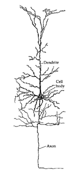
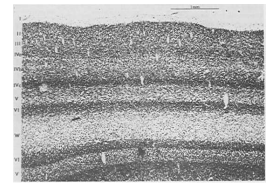
Professor: Michael Frank
Lab for Neural Computation and Cognition
`http://ski.cog.brown.edu`
`michael_frank@brown.edu`

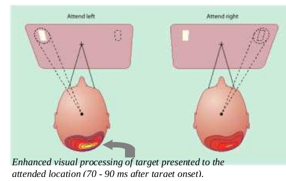
TA: Bradley Doll
`bradley_doll@brown.edu`

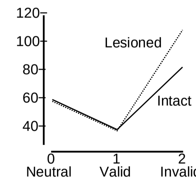
1. Syllabus; Course website; Email list.
2. Motivation for Computational Neural Models of Cognition

2

The Challenge: From Neurons to Behavior

 →  →





3

Physical Reductionism

Reductionism: explaining in terms of underlying mechanisms.

What mechanisms for cognition? CPU & RAM? Logic? Lisp? Symbolic Productions?

Physical Reductionism: mechanism is the *brain*.

→ *Look to brain itself for language and principles upon which to explain cognition.*
These principles unlikely to map onto classical box and arrow depictions of the cognitive system.

4

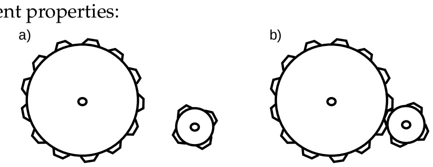
Reconstructionism

Complementary process to reductionism: Putting the reduced pieces back together.

Critical when there are billions of such pieces (neurons).

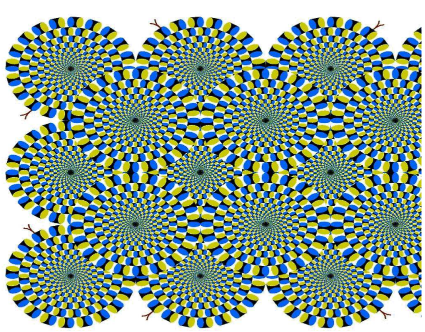
Computer simulations are essential.

Emergent properties:



5

Direct Extension to Perception...
Emergent Visual Illusion



6

Complexity and Levels of Analysis

The brain is very complex: billions of neurons, 5,000 x billion synapses, changing every nanosecond.

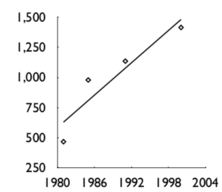
Need to abstract away from this complexity!

Is there some simpler, higher level for describing what the brain does during cognition?

7 Why should we care about computational models?

- The brain is a computing device
- Computational models can help us talk about brain functions in a precise way
- Abstract and formal theory can help us organize and interpret data

8 # pages in Principles of Neural Science (Kandel, Schwartz and Jessel)



Our task is to make this book shorter!

Describing $\neq$ Understanding

9 In a Computer Program

You can have three levels of abstraction (Marr, 1982):

Example: program that sorts lists of numbers.

Computational: *what is overall goal?*
(arrange numbers so that smallest is first in list, etc)

Algorithmic: *what strategy?*
(different sorting strategies, speed/accuracy tradeoffs).

Implementational: *how to physically encode?*
(how program written & executed using particular language)

Can we focus only on 1st two levels?

Only if you assume a particular implementation!
(emergent property of movement: gears different than joint)

10 Alternative Levels

Cognitive Phenomena

Principles

Neurobiological Mechanisms

Principles take constraints from, and are fundamentally shaped by, core levels

→ not "privileged" as in Marr's computational level

11 Neurobiological Mechanisms and Principles

Neurons: serve as *detectors*, signal with *activity*.

Networks: link, coordinate, amplify, and select patterns of activity over neurons.

Learning: organizes networks to perform *tasks* & develop *models* of environment.

Second half of course: how different brain systems solve *computational tradeoffs* imposed by conflicting cognitive demands (eg memory).

12 Psychological Phenomena Captured by Models (O'Reilly & Munakata)

Visual encoding: A network views natural scenes, and develops brain-like ways of encoding them using principles of learning.

Spatial attention: A model focuses its attention in different locations in space, and simulates performance under normal conditions and following brain damage.

Episodic memory: Replicating the structure of the *hippocampus*, a model forms new episodic memories and solves human memory tasks.

Working memory: A neural network with specialized biological (PFC) mechanisms simulates our *working memory* capacities (e.g., the ability to mentally juggle a bunch of numbers while trying to multiply 42 x 17).

13 Psychological Phenomena Captured by Models

Reinforcement learning: A network model of the *basal ganglia / dopamine* system learns to make decisions based on its experiences of positive and negative reinforcement, and reproduces patterns seen in Parkinsons patients .

Word reading: A network learns to read and pronounce nearly 3,000 English words, and *generalizes* to novel nonwords (e.g., "mave" or "nust") just like people do. Damaging a reading model simulates various forms of dyslexia.

Task directed behavior: A network simulates the "executive" part of the brain, the *prefrontal cortex*, which keeps us focused on performing the task at hand and protects us from distraction.

14 Advantages of Simulation Method

15 Potential Traps/Problems

16 Advantages of Simulation Method

- Models help us understand phenomena:

  "If you really want to understand how something works, build it."
  - Can provide novel insights.
  - Effects of brain damage/drugs.
  - Can explain *why* things are (function).
- Models deal with complexity, span levels.
- Models are explicit:
  - Deconstruct psychological constructs.
  - Make novel predictions.
  - Force accountability in simulating data.
- Enables complete control & understanding.
- Forces consistency & unity in framework.

17 Potential Traps/Problems

- Models are too simple.
- Models are too complex.
- Models can do anything:
- Models are reductionistic.

18 Multiple Levels of Computational Cognitive Models

Different levels of abstraction to account for data collected at different levels.

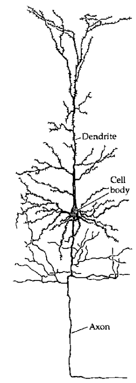
- Molecular modeling: details of neuron geometry, effects on dendritic integration, detailed ion channels
- Spiking "integrate and fire" models, less detail at single cell level, can study synchronization of firing across network of cells
- Firing rate models (Abstract connectionism); large networks of interacting neuron-like units; can approximate int and fire; brain is one big goop.
- Bayesian "optimal" approach: assumes brain performs optimal computation for any given task, usually without regard for implementation.
- Symbolic, production systems (if-then rules) *In practice these categories can blend to some degree*

19 Pros & Cons of Various Levels/Approaches

- No single one is a panacea! (as much as any researcher/lab wants/pretends for their method to be only important one..)
- Different potential for understanding cog systems vs brain
- Power for practical applications
- Biological plausibility
- Potential to inform about drugs and disease
- Number of parameters needed to fit data
- Ability to bridge gap between neurons and behavior

20 Neurons

How do they do it?

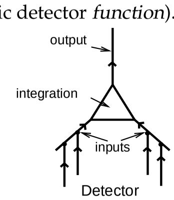


21 Detector Model

Each neuron detects some set of conditions (e.g., smoke detector).

Neurons feed on each other's outputs — layers of ever more complicated detectors.

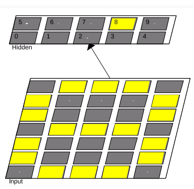
(Things can get very complex in terms of *content*, but each neuron is still carrying out basic detector *function*).



22 Networks

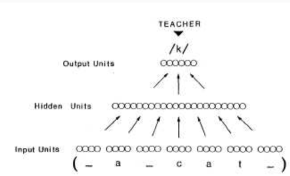
1. Biology: Cortical layers and neurons
2. Excitation:
   - Unidirectional (transformations)
   - Bidirectional (pattern completion, amplification)
3. Inhibition: Controlling bidirectional excitation.
4. Constraint Satisfaction: Putting it all together.

23 Excitation (Unidirectional): Transformations



- Detectors work in parallel to *transform* input activity pattern to hidden activity pattern.
- Emphasizes some distinctions, collapses across others.
- Function of what detectors detect (and what they ignore).

24 Application: NetTalk! (Sejnowksi & Rosenberg, 1986)



- Learns to read & pronounce english text
- Inputs are one of 29 chars (26 letters + space, comma, full stop)
- 7 letter window (provides context). total = 29x7 = 203 inputs.
- Hidden layer of 80 units.
- Output generates one of 60 phonemes, represented by 21 articulation units and 5 units for stress/syllable boundary info.

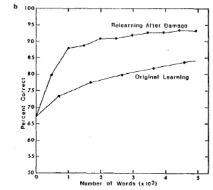
25

NetTalk: Results

- Learns regularities of english speech
- Generalizes to novel words not in training set with 78% accuracy

26

NetTalk: Results



- Knowledge is distributed: relearning after damage much faster than original training
- Distributed (spaced) practice more effective for long term retention than massed practice

27

NetTalk: Impressive, But...

- Solves reading and speaking at once (unlike people)
- Doesn't address specialization of different brain areas in language processing.
- Uses biologically implausible "error backpropagation" method for training weights
- Explicit "teacher" provides correct information on output articulatory units (instead of trial and error learning that we have to do)
- Requires many passes through exact same training set (rather than natural language experiences).

28

Extra: (Students not responsible for material on following slides)

29

Conductance models

- Single or few neurons at high detail
- Branching points of dendrites, diameters lengths of difft parts.
- Each compartment given active channels such as voltage sensitive or synaptic channels.
- Ideal for explaining spikes, thresholds for initiating spikes, precise effects of synaptic input, bursting, spike adaptation, spikes that propagate backwards up dendritic tree.
- But: huge number of parameters, only few determined from experiments.
- Complex so difficult to analyze, and virtually impossible to use for any cognitive modeling.

30

Integrate-and-fire models

- No active channels; approximation of spike generation coupled w/ leaky integrator model when voltage is below spike threshold.
- Simplified geometry, stereotyped time-course for synaptic input and spike-rate adaptation.
- Good for simulating large recurrently connected nets of neurons, synchronization and desynchronization across population, effects of noise, spike-timing dependent plasticity.
- Still pose analytical difficulties, may sometimes be unhappy medium between more realistic but analytically intractable compartmental models and highly abstract but tractable firing-rate models.

Firing rate models

- Most abstract, no spiking, outputs are continuous valued time varying firing rates.
- Approximation to int and fire, w / assumptions about time-constants of cells.
- Empirically and analytically tractable.
- Can be feedforward or recurrent.
- Coupled, non-linear diff eqn's, shows dynamical behaviors (attraction to one set of fixed points, oscillations). Regularities make them ideal as substrates for neural computation.
- Offer obvious way for large nets of simple units to perform sophisticated tasks without building knowledge in.
- Different levels of plausibility for learning rules (hebbian, error-driven, reinforcement based)