Lecture 1: Reverse engineering the brain

Samuel Gershman

Harvard University

How does the brain work?

- Asking how something works is fundamentally a question about how it serves a function.

How does the brain work?

- Asking how something works is fundamentally a question about how it serves a function.
- The heart pumps, the stomach digests, the brain thinks.

How does the brain work?

- Asking how something works is fundamentally a question about how it serves a function.
- The heart pumps, the stomach digests, the brain thinks.
- The real question: *How does the brain produce thought?*

How does the brain work?

- Asking how something works is fundamentally a question about how it serves a function.
- The heart pumps, the stomach digests, the brain thinks.
- The real question: *How does the brain produce thought?*
- **Thought is computation**—the manipulation of representations for some purpose.

Representations

- A collection of neurons represents something (e.g., an apple) in the sense that a downstream neuron can interpret their activity pattern in terms of information about the apple.

Representations

- A collection of neurons represents something (e.g., an apple) in the sense that a downstream neuron can interpret their activity pattern in terms of information about the apple.
- The downstream neuron can then do something with this information by participating in a computation (e.g., planning a reaching movement, comparing the apple to other apples in memory, deciding whether to eat it, etc.).

Representations

- A collection of neurons represents something (e.g., an apple) in the sense that a downstream neuron can interpret their activity pattern in terms of information about the apple.
- The downstream neuron can then do something with this information by participating in a computation (e.g., planning a reaching movement, comparing the apple to other apples in memory, deciding whether to eat it, etc.).
- Mental computations are purposeful manipulations of representations.

Marr's levels

We will try to understand what constitutes "purposeful manipulations of representations" at multiple levels of analysis.

Marr's levels

We will try to understand what constitutes "purposeful manipulations of representations" at multiple levels of analysis.

1. **Computational level**: What is the problem being solved by the system?

Marr's levels

We will try to understand what constitutes "purposeful manipulations of representations" at multiple levels of analysis.

1. **Computational level**: What is the problem being solved by the system?
2. **Representational/algorithmic level**: How is the problem solved algorithmically?

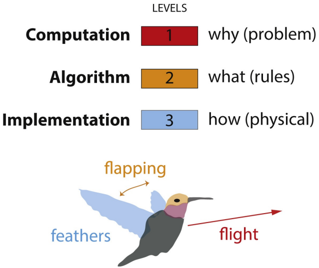
Marr's levels

We will try to understand what constitutes "purposeful manipulations of representations" at multiple levels of analysis.

1. **Computational level**: What is the problem being solved by the system?
2. **Representational/algorithmic level**: How is the problem solved algorithmically?
3. **Implementation level**: How is the algorithm realized physically?

Why are the levels useful?

Marr: "Trying to understand perception by studying only neurons is like trying to understand bird flight by studying only feathers: It just cannot be done."



[Krakauer et al. 2017]

Why are the levels useful?

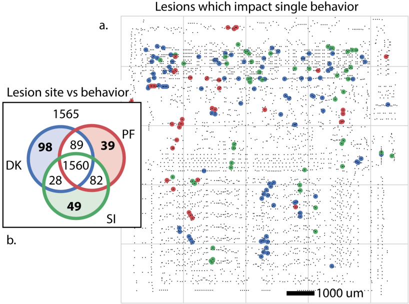
1. **Conceptual**: nothing recognizable as "cognition" if one only looks at neurons.

Why are the levels useful?

1. **Conceptual**: nothing recognizable as “cognition” if one only looks at neurons.
2. **Methodological**: Thinking like an engineer is often a good starting place for building models.

How far can we go with a hardcore bottom-up approach?

Lesioning a microprocessor to identify the functions of individual transistors: an exercise in futility?



[Jonas & Kording 2017]

The reverse engineering approach

1. Posit the computational problems that the brain is trying to solve.

The reverse engineering approach

1. Posit the computational problems that the brain is trying to solve.
2. Engineer algorithmic solutions to these problems.

The reverse engineering approach

1. Posit the computational problems that the brain is trying to solve.
2. Engineer algorithmic solutions to these problems.
3. Model how the brain could implement the algorithmic solutions under biological constraints.

Study question

What are the advantages and disadvantages of the reverse engineering approach?

The computational level: statistical decision problems

- At first glance, the brain seems to be solving many different kinds of problems (perception, decision making, motor control, learning, memory, language understanding, etc.).

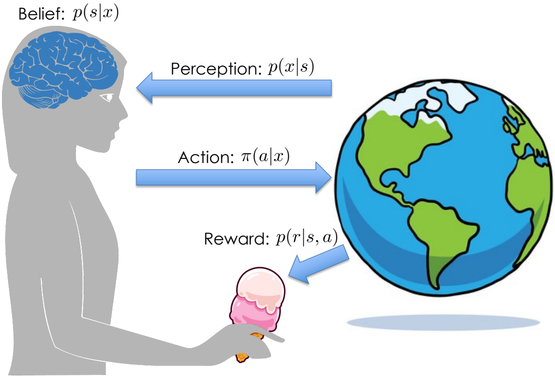
The computational level: statistical decision problems

- At first glance, the brain seems to be solving many different kinds of problems (perception, decision making, motor control, learning, memory, language understanding, etc.).
- On further inspection, these all appear to be variations on one kind of problem: a statistical decision problem.

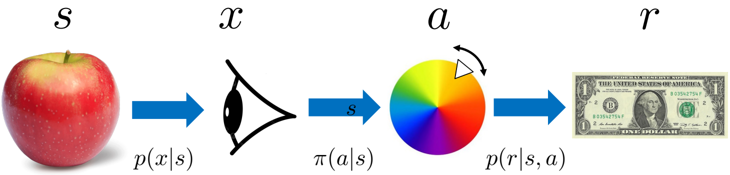
The computational level: statistical decision problems

- At first glance, the brain seems to be solving many different kinds of problems (perception, decision making, motor control, learning, memory, language understanding, etc.).
- On further inspection, these all appear to be variations on one kind of problem: a statistical decision problem.
- This is general enough to encompass many (all?) specific functions carried out by the brain.

Decision theory



Example



Utility

- The agent gets utility $u(r)$ from reward r.

Utility

- The agent gets utility $u(r)$ from reward r.
- For example, r might be money you earn from the task I give you, and $u(r)$ is how much you value the money, which depends on factors like your current wealth level and the purchasing power of the money.

Utility

- The agent gets utility $u(r)$ from reward r.
- For example, r might be money you earn from the task I give you, and $u(r)$ is how much you value the money, which depends on factors like your current wealth level and the purchasing power of the money.
- This emphasizes the fact that utility is distinct from nominal quantities like dollars, number of calories, etc. Utility is internally generated.

Bayesian decision theory

- Key idea: maximize *expected utility* $\bar{u}(\pi) = \mathbb{E}[u(r)|\pi]$ given beliefs $p(s|x)$ about the hidden state of the world:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\ \bar{u}(\pi).$$

Bayesian decision theory

- Key idea: maximize *expected utility* $\bar{u}(\pi) = \mathbb{E}[u(r)|\pi]$ given beliefs $p(s|x)$ about the hidden state of the world:

$$\pi^* = \operatorname*{argmax}_{\pi} \bar{u}(\pi).$$

- Expected utility is how much utility an agent believes it will gain under policy π, averaging over these sources of randomness:

$$\bar{u}(\pi) = \sum_x p(x) \sum_a \pi(a|x) \sum_s p(s|x) \sum_r p(r|s,a)u(r).$$

Bayesian inference

- The posterior $p(s|x)$ is the agent's probabilistic belief about the hidden state s given the signal x.

Bayesian inference

- The posterior $p(s|x)$ is the agent's probabilistic belief about the hidden state s given the signal x.
- Update from prior $p(s)$ to posterior according to Bayes' rule:

$$p(s|x) = \frac{p(x|s)p(s)}{p(x)}.$$

Bayesian inference

- The posterior $p(s|x)$ is the agent's probabilistic belief about the hidden state s given the signal x.
- Update from prior $p(s)$ to posterior according to Bayes' rule:

$$p(s|x) = \frac{p(x|s)p(s)}{p(x)}.$$

- The likelihood, $p(x|s)$, expresses how well a hypothetical state "fits" the data.

Study question

If priors are subjective, are Bayesian theories unfalsifiable?

Why be Bayesian?

The logician's argument

Why be Bayesian?

The logician's argument

- In Boolean logic, the truth value of a proposition is represented by 0 (false) or 1 (true)

Why be Bayesian?

The logician's argument

- In Boolean logic, the truth value of a proposition is represented by 0 (false) or 1 (true)
- The truth value of a complex propositions can be calculated using Boolean algebra.
  - Conjunction: AB
  - Disjunction $A + B - AB$
  - Negation: $1 - A$

Why be Bayesian?

The logician's argument

- In Boolean logic, the truth value of a proposition is represented by 0 (false) or 1 (true)
- The truth value of a complex propositions can be calculated using Boolean algebra.
  - Conjunction: AB
  - Disjunction $A + B - AB$
  - Negation: $1 - A$
- Truth values are known with certainty (Boolean operations always yield values of 0 or 1). What if you're unsure? Is there a "soft" version that correctly represents and propagates a measure of "plausibility"?

Why be Bayesian?

The logician's argument

- We'd like "plausibility" to be a real number (so that it can encode continuous degrees of plausibility) and internally consistent (logically equivalent propositions should have the same plausibility).

Why be Bayesian?

The logician's argument

- We'd like "plausibility" to be a real number (so that it can encode continuous degrees of plausibility) and internally consistent (logically equivalent propositions should have the same plausibility).
- We'd also like it to recover Boolean logic as a special case when plausibility is maximal or minimal (corresponding to complete certainty).

Why be Bayesian?

The logician's argument

- We'd like "plausibility" to be a real number (so that it can encode continuous degrees of plausibility) and internally consistent (logically equivalent propositions should have the same plausibility).
- We'd also like it to recover Boolean logic as a special case when plausibility is maximal or minimal (corresponding to complete certainty).
- Cox's Theorem: only probabilities updated according to Bayes' rule satisfy these requirements.

Why be Bayesian?

The logician's argument

- We'd like "plausibility" to be a real number (so that it can encode continuous degrees of plausibility) and internally consistent (logically equivalent propositions should have the same plausibility).
- We'd also like it to recover Boolean logic as a special case when plausibility is maximal or minimal (corresponding to complete certainty).
- Cox's Theorem: only probabilities updated according to Bayes' rule satisfy these requirements.
- Thus, Bayesian probability theory can be viewed as a natural extension of Boolean logic.

Why be Bayesian?

The decision theorist's argument

Why be Bayesian?

The decision theorist's argument

- Admissible policy: at least as good as any other policy across all states.

Why be Bayesian?

The decision theorist's argument

- Admissible policy: at least as good as any other policy across all states.
- Complete Class Theorem: every admissible policy corresponds to a Bayesian policy for some prior.

Why be Bayesian?

The decision theorist's argument

- Admissible policy: at least as good as any other policy across all states.
- Complete Class Theorem: every admissible policy corresponds to a Bayesian policy for some prior.
- Thus, Bayesian decision theory is in a sense inevitable for a decision maker who wants to avoid being dominated.

Why be Bayesian?

The gambler's argument

Why be Bayesian?

The gambler's argument

- Dutch Book Theorem: a bookie can't make money off of a Bayesian agent in expectation.

Why be Bayesian?

The gambler's argument

- Dutch Book Theorem: a bookie can't make money off of a Bayesian agent in expectation.
- If instead the agent makes bets using plausibilities that violate the axioms of probability, then it's possible to construct a bet that they will accept and yet they will be guaranteed to lose money.

Why be Bayesian?

The gambler's argument

- Dutch Book Theorem: a bookie can't make money off of a Bayesian agent in expectation.
- If instead the agent makes bets using plausibilities that violate the axioms of probability, then it's possible to construct a bet that they will accept and yet they will be guaranteed to lose money.
- Thus, there is a financial incentive to be Bayesian.

The algorithmic level

- Real agents have constraints on computation, memory, and data.

The algorithmic level

- Real agents have constraints on computation, memory, and data.
- These constraints delimit what kinds of algorithms are realizable.

Complexity, efficiency, tractability

We can characterize the requirements of an algorithm along several dimensions:

1. **Time complexity**: how much computation is required?
2. **Space complexity**: how much memory is required?
3. **Sample complexity**: how much data are required?

If complexity cannot be expressed as a polynomial function of the input size N, an algorithm is considered inefficient. A problem for which no efficient algorithm exists is intractable.

Illustration

- Suppose the state space consists of N variables, $s = (s_1, \ldots, s_N)$, where each variable can take one of K discrete values.

Illustration

- Suppose the state space consists of N variables, $s = (s_1, \dots, s_N)$, where each variable can take one of K discrete values.
- Computing the normalizing constant for Bayes' rule then requires summing over K^N possible configurations.

Illustration

- Suppose the state space consists of N variables, $s = (s_1, \ldots, s_N)$, where each variable can take one of K discrete values.
- Computing the normalizing constant for Bayes' rule then requires summing over K^N possible configurations.
- Example: x corresponds to images and s corresponds to the set of N objects in a scene, each of which could belong to K possible categories.

Illustration

- Suppose the state space consists of N variables, $s = (s_1, \ldots, s_N)$, where each variable can take one of K discrete values.
- Computing the normalizing constant for Bayes' rule then requires summing over K^N possible configurations.
- Example: x corresponds to images and s corresponds to the set of N objects in a scene, each of which could belong to K possible categories.
- Enumerating all possible states is inefficient because N appears in the exponent.

Combinatorial problems

- Combinatorial problems, with complexity exponential in N, are everywhere.

Combinatorial problems

- Combinatorial problems, with complexity exponential in N, are everywhere.
- They cannot be efficiently solved by algorithms that rely on exhaustive enumeration.

Combinatorial problems

- Combinatorial problems, with complexity exponential in N, are everywhere.
- They cannot be efficiently solved by algorithms that rely on exhaustive enumeration.
- Exponential complexity frequently arises in high-dimensional problems where some computation requires exhaustive coverage of the space—the *curse of dimensionality*.

Tractability and the brain

- If a problem is intractable, it is unlikely that our brains evolved to solve it. This suggests the following research strategy:

Tractability and the brain

- If a problem is intractable, it is unlikely that our brains evolved to solve it. This suggests the following research strategy:
- Only reverse engineer the brain's efficient solutions to tractable problems.

Tractability and the brain

- If a problem is intractable, it is unlikely that our brains evolved to solve it. This suggests the following research strategy:
- Only reverse engineer the brain's efficient solutions to tractable problems.
- Focus on algorithms with polynomial complexity that have been shown to work in practice.

Tractability and the brain

- If a problem is intractable, it is unlikely that our brains evolved to solve it. This suggests the following research strategy:
- Only reverse engineer the brain's efficient solutions to tractable problems.
- Focus on algorithms with polynomial complexity that have been shown to work in practice.
- Identify behavioral and neural signatures of these algorithms, investigate how they could be implemented with neural machinery.

Resource rationality

- Each agent has a capacity limit, $\mathcal{C}$, measured in resource units (e.g., time, memory, computation, information).

Resource rationality

- Each agent has a capacity limit, $\mathcal{C}$, measured in resource units (e.g., time, memory, computation, information).
- The resource-rational policy optimizes expected utility subject to the resource capacity limit:

$$\pi^* = \operatorname*{argmax}_{\pi:\, c(\pi) \leq \mathcal{C}} \bar{u}(\pi).$$

where $c(\pi)$ is the amount of resources consumed by implementing policy π.

Study question

How is it possible to find optimal resource-constrained policies when the policy search is itself resource-constrained? And doesn't this threaten an infinite regress, where each optimization is nested within an even more difficult optimization problem?

The implementation level

- There are many physical implementations of a given algorithm.

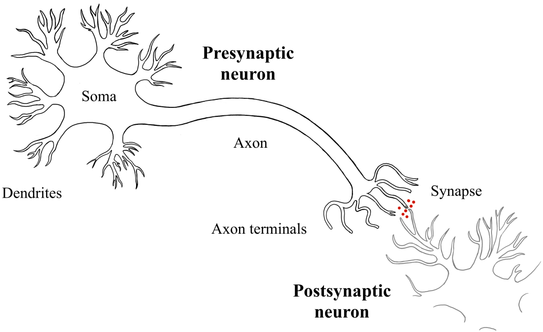
The implementation level

- There are many physical implementations of a given algorithm.
- The brain's elementary computing units are neurons.

The implementation level

- There are many physical implementations of a given algorithm.
- The brain's elementary computing units are neurons.
- Each neuron implements a relatively simple computation; wiring up many neurons together makes complex computation possible.

The implementation level



Resource-rational neurobiology

- Spikes are metabolically expensive; cells cannot spike with arbitrarily high rates for arbitrarily long periods of time.

Resource-rational neurobiology

- Spikes are metabolically expensive; cells cannot spike with arbitrarily high rates for arbitrarily long periods of time.
- Maintaining a reliable response to inputs is also metabolically expensive.

Resource-rational neurobiology

- Spikes are metabolically expensive; cells cannot spike with arbitrarily high rates for arbitrarily long periods of time.
- Maintaining a reliable response to inputs is also metabolically expensive.
- Maintenance of synaptic weights is metabolically expensive.

Resource-rational neurobiology

- Spikes are metabolically expensive; cells cannot spike with arbitrarily high rates for arbitrarily long periods of time.
- Maintaining a reliable response to inputs is also metabolically expensive.
- Maintenance of synaptic weights is metabolically expensive.
- The brain should economize on the number of neurons, their average firing rate, the reliability of firing, and the number/strength of connections between neurons.

Summary

- The brain implements thought. Thought is computation (purposeful manipulation of representations).

Summary

- The brain implements thought. Thought is computation (purposeful manipulation of representations).
- The purpose: Bayesian decision theory.

Summary

- The brain implements thought. Thought is computation (purposeful manipulation of representations).
- The purpose: Bayesian decision theory.
- The representations: probabilistic beliefs, utilities, and costs.

Summary

- The brain implements thought. Thought is computation (purposeful manipulation of representations).
- The purpose: Bayesian decision theory.
- The representations: probabilistic beliefs, utilities, and costs.
- The implementational primitives: networks of neurons connected by plastic synapses.