1.2 Memory as knowledge: semantic memory

Maintaining knowledge of the environment is a crucial responsibility of human memory, and this responsibility rests primarily with the semantic memory system. This section aims to provide an overview of key ideas from prior research that can serve as a foundation for a normative account of how this responsibility can be met. First, we will outline the reasons for why knowledge representation in the brain, particularly in semantic memory, should be in the form of a probabilistic generative model. Then, we briefly introduce the theoretical foundations of probabilistic models, including graphical models and probabilistic programs. We show how this approach addresses many of the key challenges in cognition, including perception as unconscious inference. Finally, we examine how learning, or the construction of a representation of the environment over an organism's lifetime, can be achieved within the framework of probabilistic inference.

1.2.1 Knowledge representation

How can knowledge be represented in a physical system such as the brain? We have argued that to perform predictions, complex organisms shape part of their brain into an artefact that in some sense mirrors the environment. Human-made prediction devices, for example, orreries such as the Antikythera mechanism serve as intuitive examples of this notion of mirroring. The Antikythera mechanism is an ancient device from the 1st or 2nd century BCE and is considered one of the first computers. The device had multiple dials representing observable astronomical information including the positions of the Sun and Moon, the moon phase, eclipses, and possibly the locations of planets. By turning the hand crank on the side of the device, interlocking gears within the mechanism moved the dials to reflect a future state of celestial objects, allowing the operator to predict future events, such as the date of the next eclipse, by reading off the dials. The Antikythera device is a physical system, parts of which (dial positions) map onto important features of another physical system (celestial body locations), and crucially, these mappings are preserved under certain transformations (e.g. time evolution). While a precise definition of the concepts of representation and computation are difficult ques-

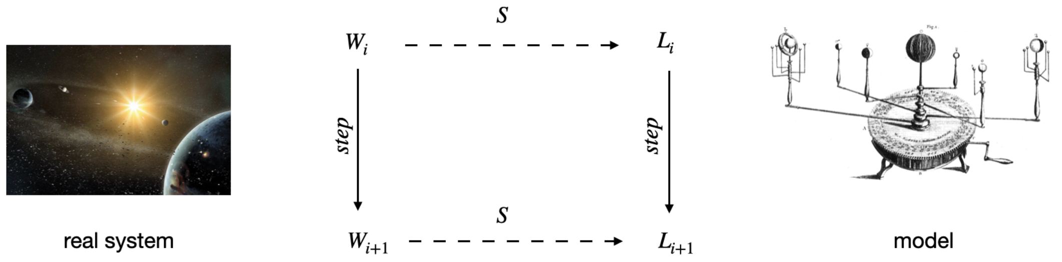

Figure 1.4: Representing knowledge of the environment in physical systems. The orrery on the right serves as a model of the solar system. The states of the real system, W_i, correspond to representational states of the orrery, L_i, and this correspondence, S, is maintained over the time evolution of the systems, allowing the user to predict the relative locations of the celestial bodies in the future.

tions [12, 13], this example intuitively illustrates how a physical system can represent knowledge regarding an external physical system (Fig. 1.4).

Note that to be useful for prediction, the variables on the dials had to transform in the same way as the positions of the celestial bodies, even though the mechanism in the former case was a system of gears whereas in the former it was the force of gravity between planets and stars. This raises the important question of what are the limits of such representation. Which kinds of mechanisms can model which other kinds of mechanisms? Perhaps most importantly from our point of view, which kinds of systems can be modelled by neurons? One of the most important discoveries of the 20th century is that surprisingly, all ways of defining effective computation led to the same set of functions. The definition that most closely aligns with computation with physical devices was developed by Turing, using abstract machines that process discrete symbols arranged on an infinite tape using a set of defined instructions known as Turing machines. Inspired by his work, McCulloch and Pitts [14] analysed idealised neural networks and suggested that recurrent versions of these networks coupled with memory could calculate the same functions as Turing machines. Although this was just a conjecture [3], such a construction was developed much later by others [16]. Turing also demonstrated that it was possible to build a universal Turing machine (UTM) that could simulate any other Turing machine by encoding its instructions on the tape. These results led to the formulation of the Church-Turing thesis, which asserts that any in-

[3] It is a common misconception that they have proved this, for details see Piccinini, 2020 [15].

tuitively computable function is computable by some TM and consequently by any other sufficiently intricate (Turing complete) mechanism. This notion of *computational universality*, the independence of the computation from the concrete physical mechanism is the basis for the existence of software separately from its hardware level implementation. Furthermore, it suggests that the brain being made of neurons can be consistent with it performing computations that are easier to understand in formal systems adapted to the high-level challenges that it faces, providing the basis for a top-down analysis.

Logic

The universality of computation suggests that we may begin our analysis in a framework ideally suited to the problems of representing knowledge in a format that can support predictions, simulations and reasoning. We start with logic, the system that describes the laws of sound reasoning that originated with Aristotle and was refined by many others [17, 18]. Specifically, we introduce a simple variant called propositional logic. To construct a model in propositional logic, we have to specify the distinct states of the system. We can decompose these states using state variables (atomic propositions). The full model can be encoded as a truth table, where columns represent the atomic variables in the system and the rows list all possible combinations of values for those variables. Each row in the truth table represents a state of the system, with the values for the atomic variables given in the corresponding columns. The last column of the truth table defines whether the combination of truth values represented in the row is a possible state of the system (see Fig 1. for an example).

In order to be able to make inferences in a logical system represented by a truth table, we can simply cross out all states that do not match the query or are not possible. For example, if we make the query: '*If the patient is coughing but doesn't have the flu, is it TB?*', we need to cross out all states where $cough \neq 1$ or $flu \neq 0$ and all states where $possible \neq 1$ (Fig. 1.5). The only possible state left is row 3, where $TB = 1$ and therefore the answer to the question in this simple model is yes.

The first issue with this method becomes apparent if we start refining the model by adding new variables: the number of rows scales exponentially with

Cough	Flu	TB	Possible
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	1

Cough	Flu	TB	Possible
~~1~~	~~1~~	~~1~~	~~1~~
~~1~~	~~1~~	~~0~~	~~1~~
1	0	1	1
~~1~~	~~0~~	~~0~~	~~0~~
~~0~~	~~1~~	~~1~~	~~0~~
~~0~~	~~1~~	~~0~~	~~0~~
~~0~~	~~0~~	~~1~~	~~0~~
~~0~~	~~0~~	~~0~~	~~1~~

Figure 1.5: Truth tables and inference in a toy diagnostic model. **Left,** the original truth table. **Right,** the truth table with lines incompatible with the query '*If the patient is coughing but doesn't have the flu, is it TB?*' crossed out.

the number of variables. This makes both representing the table and answering queries very cumbersome and therefore it is useful to introduce logical operators. Each operator is defined by a smaller table called a conditional truth table (Fig. 1.6). Using these CTTs to decompose the full truth table makes the definition of the model much shorter. For example, using logical operators, the truth table in Fig 1. can be described simply by:

$$(\text{flu} \ \vee \ \text{TB}) \ \leftrightarrow \ \text{cough}$$

Moreover, these operators enable more efficient means of performing inference than the method of crossing out lines in the truth table:

$$A \leftrightarrow B, A \vdash B$$

$$A \vee B, \neg A \vdash B$$

For example, to answer the question '*If the patient is coughing but doesn't have the flu, is it TB?*', or in logical notation:

$$\text{cough} \wedge \neg\text{flu} \vdash \text{TB},$$

we can use the above two inference rules by substituting into the first rule,

$$\text{cough} \leftrightarrow (\text{flu} \ \vee \ \text{TB}) \ , \text{cough} \vdash (\text{flu} \ \vee \ \text{TB}),$$

and then substituting the above result into the second inference rule, we get:

$$\text{flu} \vee \text{TB}, \neg\text{flu} \vdash \text{TB}.$$

A	B	A $\leftrightarrow$ B
1	1	1
1	0	0
0	1	0
0	0	1

A	B	A $\vee$ B
1	1	1
1	0	1
0	1	1
0	0	0

Figure 1.6: Conditional truth tables for the IFF operator (A $\leftrightarrow$ B) and the OR operator (A $\vee$ B).

Therefore, we can conclude that in this situation, TB is true.

Despite the introduction of these logical operators, the expressive power of propositional logic remains limited. However, let us consider a different kind of problem first. Assume the question was '*If the patient is coughing, does he have TB?*'. If we attempt to use the method of crossing out incompatible and impossible states of the world, we are left with only three possible states: two in which the patient has TB and one in which he does not. This means that there are queries which are undecidable in the system and in such cases, logic fails to provide a useful answer.

Undecidability and degrees of belief

Unfortunately in practical situations, the majority of queries that the agent is concerned with will fall into the category of undecidable or uncertain. However, there is a simple modification we can make to allow the system to deal with uncertain knowledge: we can allow the 'possible' column to take on non-binary values ranging from zero to one, corresponding to the agent's degree of belief or plausibility that the world is in that state. How should the agent choose these numbers? Two well-known arguments exist regarding how degrees of belief should be consistently assigned to states of the world.

First, Cox [19] attempted to demonstrate that if an agent's degrees of belief are expressed as real numbers and satisfy axioms encoding common sense requirements, such as the requirement that different ways of arriving at an answer from the same information should lead to the same outcome, then these beliefs must adhere to the axioms of probability theory. Therefore, in extending logic to handle uncertainty, the degrees of belief should be chosen such that they satisfy the rules of classical probability theory. Cox's theorem can be seen alternative axiomatisation of probability theory, equivalent to Kolmogorov's axioms but with a different motivation. In this correspondence with probability theory, queries are formalised as the calculation of conditional probabilities, where the condition defines the query. For example, our query in the diagnosis system would correspond to the computation of $P(\text{TB}|\text{cough}, \neg\text{flu})$. It can be shown that the computation of conditional probabilities is equivalent to the method of crossing out the incompatible rows of the truth table and renormalising the remaining rows to sum to unity.

The second argument, called the *Dutch book argument* [20], assumes that the agent is willing to make bets according to his degrees of belief. The argument shows that if these degrees of belief fail to follow certain consistency rules, there will exist a set of bets that the agent would accept as fair, even though it would guarantee a loss. Such a set of bets is called a 'dutch book' by professional bookmakers. As in Cox's theorem, the consistency rules that degrees of belief have to satisfy correspond to the rules of classical probability theory.

Taken together, these two arguments are key pieces in motivating the use of probability theory as a normative framework for understanding human cognition. In particular, they suggest that the agent's internal models should be formalised as probabilistic models of the environment with the probabilities corresponding to how plausible the agent considers each state to be. The agent can query its representation through computing conditional probabilities, which corresponds to probabilistic inference. We will discuss in more detail how various cognitive functions can be cast as probabilistic inference in section 1.2.2, but first, we return to the problem of expressivity.

Graphical models

Joint probability tables, the probabilistic extensions of truth tables suffer from the same '*curse of dimensionality*' as their counterparts from logic, that is, the problem of exponential scaling with the number of variables. Similarly to logic, there are extensions of probability theory that exploit compositionality to increase the expressivity of probability tables. Analogously to the way that truth tables can be decomposed using operators defined by conditional truth tables, joint probability tables can be decomposed using smaller conditional probability tables. This trick leads to the concept of directed graphical models in probability theory. In graphical models, each variable in the model corresponds to a node in a directed acyclic graph (DAG), where the parents of each node are the variables which are included in the conditional probability table for the variable represented by the node. In addition to allowing for a more compact representation of the joint probability table, graphical models also enable inference algorithms that make certain kinds of queries such as statistical dependencies between the variables more efficient to compute.

Since graphical models play a central role in probabilistic models of cognition, we consider them in more detail. In order to see how graphical models enable a more compact representation of the joint distribution, we first use the chain rule for probabilities to factorise it for an arbitrary ordering $X_1, X_2, .., X_n$ of n random variables as

$$P(x_1, x_2, \ldots, x_n) = P(x_1|x_2, \ldots, x_n)P(x_2|x_3, \ldots, x_n) \ldots P(x_n).$$

If the conditional probability of a certain variable X_i is only dependent on a subset of the variables that it is conditioned on, and we call this subset pa_i (the Markovian parents of X_i) then we have

$$P(x_i|x_{i+1}, ..., x_n) = P(x_i|pa_i).$$

This means that the joint distribution can be written as

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_I|pa_i),$$

which can greatly reduce the information needed to specify it (that would otherwise require 2^n entries even for binary variables) by decomposing it into smaller distributions. Furthermore, we can construct a DAG that satisfies the same child-parent relationships with each variable having a corresponding node and each conditional dependence a directed edge. We call such a graph G a graph representation of distribution P. We introduce the following notation for independence between variables

$$X \perp Y \quad \Leftrightarrow \quad P(x) = P(x|y) \quad \Leftrightarrow \quad P(x, y) = P(x)P(y),$$

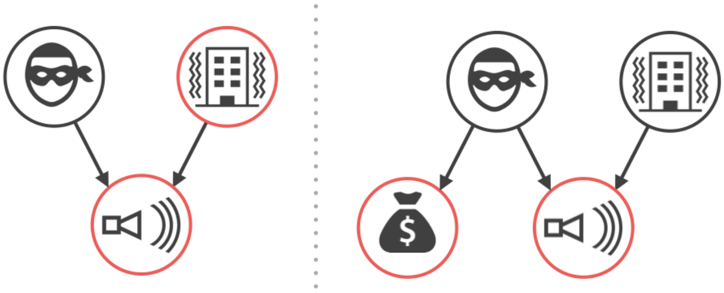

Figure 1.7: An illustration of the explaining away effect for probabilistic models. **Left:** We suppose that burglaries and earthquakes are unrelated and rare, but both cause the house alarm to signal. It follows that being informed that the alarm went off increases our concern about both an earthquake and a burglary being in the house. However, if, in addition to hearing the alarm, we also feel the ground trembling, it makes us less likely to think that we have also been the victims of burglary. **Right:** Direct observation of latents is not necessary for them to become dependent - if our model also includes the fact that valuables go missing after a break-in, then noticing both the alarm going off and the disappearance of possessions is enough to infer the presence of the thief and conclude that there was probably no earthquake.

and for conditional dependence

$$X \perp Y|Z \quad \Leftrightarrow \quad P(x|z) = P(x|y,z) \quad \Leftrightarrow \quad P(x,y|z) = P(x|z)P(y|z).$$

The graph representation provides an economical encoding of the independence relations or correlation structure of the distribution. It also enables these relations to be read out via graph-search algorithms instead of algebraic methods, and can be given a causal interpretation. We say that probabilistic influence can flow from variable X to variable Y through a set of variables Z if $X \not\perp Y|Z$. Independencies can be read off the graph representation by checking whether there exists a trail (a path where the edges can point in either direction) through which influence can flow between the variables in question [21]. The flow of influence is blocked by conditioning on nodes Z if an only if:

1. the trail contains $u \leftarrow m \leftarrow v$, such that $m \in Z$,
2. the trail contains $u \leftarrow m \rightarrow v$, such that $m \in Z$,
3. or the trail contains $u \rightarrow m \leftarrow v$ such that $m \notin Z$ and no descendants of m are in Z,

where u, m and v are nodes in graph G. An interesting dependence relation is made evident by these criteria, called selection bias in the statistical and *explaining away* in the AI literature. The third criteria implies that otherwise independent parents can become correlated if we condition on their mutual child. Informally, if we observe something that can be a cause of some other observation, then this reduces the need for an alternative explanation. This also happens if we don't observe the hidden cause, but infer it from some auxiliary observation. An illustration of this effect can be found in Fig 1.7.

While both the introduction of operators and inference rules in propositional logic and graphical models in probability theory are important conceptual steps in dealing with the curse of dimensionality of truth tables and probability tables respectively, their expressive power is still limited. Logic has been extended into higher order logics, for example through the introduction of predicates and quantifiers results in first order logic. An important development was λ-calculus, which in addition to corresponding to a higher-order logic, has equivalent expressive power to universal Turing machines and thereby capable of expressing any effectively computable function. λ-calculus can be directly extended to handle probabilities with the introduction of a random sampling operator. This extension, called stochastic λ-calculus, provides the basis for a Turing-complete formalisation of probabilistic models called probabilistic programs [22].

Probabilistic programs

Probabilistic programs provide a Turing-complete modelling language that allows for the construction of mental simulations of any computable generative process. Each program represents a probability distribution through the relative frequencies of its outputs when executed. Running the program simulates the environment and results in a possible world state that is consistent with the model's initial conditions. Each of these output states can be considered a sample from the distribution represented by the program and as the number of samples grows, the relative frequencies of the outputs tend towards their probabilities, providing a sampling or Monte Carlo representation (for more details on sampling representations see section 1.3.1).

Similarly to directed graphical models, an important advantage of probabilistic programs that they can represent causal models. They have been proposed as a framework for formalising core human competencies such as intuitive physics and intuitive psychology by enabling the rich mental simulations of causal processes that these require [23]. For example, intuitive physics has been proposed to be similar to physics engines in modern video games that are used to create environments for the players to interact with. These physics engines contain simplified objects with properties and approximate dynamics for updating the world state and usually also incorporate a graphics engine that renders the 3D visual scene from the perspective of the player. This hypothesis has been successfully applied in a range of human experiments [24, 25]. In relation to this thesis, probabilistic programs are important primarily as a general formalisation of the compositionally defined, open-ended hypothesis spaces that the human brain has to be able to navigate during learning. We explore this view of learning as a process of finding the probabilistic program that generates the observed data in section 1.2.3.

1.2.2 Probabilistic models of cognition

As discussed in the previous section, making inferences is a key part of knowledge representation. Probabilistic inference has emerged as a unifying language for modelling inference problems that arise in the fields of cognitive science, machine learning and computational neuroscience. In this section, we provide an overview of how important cognitive functions, such as decision-making and perception, can be framed as probabilistic inference, supported by the world model stored in semantic memory.

Perception as inference

A fundamental challenge for the brain is the difference between the quantities it can directly measure through sensory neurons and those that are relevant to plans and decisions. Quantities in the former category are things like impacts of photons, vibrations in surrounding air, temperature and the presence of certain molecules, whereas the brain is more concerned with determining the presence and properties of objects, animals, individuals, and their thoughts and intentions. This process of inferring the latent quantities based on the directly measurable ones was termed unconscious inference by Helmholtz [26], alluding to the observation that introspectively we only have access to the result of

References

[1] John R. Anderson. 'Is human cognition adaptive?' In: *Behavioral and Brain Sciences* 14.3 (1991), pp. 471–485. DOI: 10.1017/S0140525X00070801.

[2] D. Marr and T. Poggio. 'From Understanding Computation to Understanding Neural Circuitry'. In: (May 1976).

[3] James L. Mcclelland, David E. Rumelhart, and PDP Research Group. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. en. MIT Press, July 1987.

[4] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. en. MIT Press, Feb. 1986.

[5] Jeff Hawkins and Sandra Blakeslee. *On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines*. en. Macmillan, Apr. 2007.

[6] William Bialek and Naftali Tishby. 'Predictive Information'. In: *Arxiv* (1999).

[7] Beren Millidge. 'Towards a Mathematical Theory of Abstraction'. In: *arXiv:2106.01826 [cs, stat]* (June 2021).

[8] Alexander A. Alemi. 'Variational Predictive Information Bottleneck'. In: *arXiv preprint* (2019), pp. 1–6.

[9] Szabolcs Káli and Peter Dayan. 'Off-line replay maintains declarative memories in a model of hippocampal-neocortical interactions'. In: *Nature Neuroscience* 7.3 (2004), pp. 286–294. DOI: 10.1038/nn1202.

[10] Pernille Hemmer and Mark Steyvers. 'Integrating episodic and semantic information in memory for natural scenes'. In: *Proceedings 31st Annual Meeting of the Cognitive Science Society* (2009), pp. 1557–1562.

[11] Pernille Hemmer and Mark Steyvers. 'A Bayesian Account of Reconstructive Memory'. In: *Topics in Cognitive Science* 1.1 (2009), pp. 189–202. DOI: 10.1111/j.1756-8765.2008.01010.x.

[12] Gualtiero Piccinini and Corey Maley. 'Computation in Physical Systems'. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2021. Metaphysics Research Lab, Stanford University, 2021.

[13] László E. Szabó. 'Mathematical Facts in a Physicalist Ontology'. In: *Parallel Processing Letters* 22.03 (Sept. 2012), p. 1240009. DOI: `10.1142/S0129626412400099`.

[14] Warren S. McCulloch and Walter Pitts. 'A logical calculus of the ideas immanent in nervous activity'. en. In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. DOI: `10.1007/BF02478259`.

[15] Gualtiero Piccinini. 'The First Computational Theory of Cognition: McCulloch and Pitts's "A Logical Calculus of the Ideas Immanent in Nervous Activity"'. In: *Neurocognitive Mechanisms: Explaining Biological Cognition.* Ed. by Gualtiero Piccinini. Oxford University Press, Nov. 2020, p. 0. DOI: `10.1093/oso/9780198866282.003.0006`.

[16] H. T. Siegelmann and E. D. Sontag. 'On the Computational Power of Neural Nets'. en. In: *Journal of Computer and System Sciences* 50.1 (Feb. 1995), pp. 132–150. DOI: `10.1006/jcss.1995.1013`.

[17] George Boole. *The Laws of Thought (1854)*. en. Open court publishing Company, 1854.

[18] E. T. Jaynes. *Probability Theory: The Logic of Science*. en. Cambridge University Press, 1979.

[19] R. T. Cox. 'Probability, Frequency and Reasonable Expectation'. In: *American Journal of Physics* 14.1 (Jan. 1946), pp. 1–13. DOI: `10.1119/1.1990764`.

[20] R. Sherman Lehman. 'On confirmation and rational betting'. en. In: *The Journal of Symbolic Logic* 20.3 (Sept. 1955), pp. 251–262. DOI: `10.2307/2268221`.

[21] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. en. MIT Press, July 2009.

[22] Noah D. Goodman et al. 'Church: a language for generative models'. In: (June 2012), pp. 220–229.

[23] Tomer D Ullman and Joshua B Tenenbaum. 'Bayesian Models of Conceptual Development: Learning as Building Models of the World'. en. In: (2020).

[24] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. 'Simulation as an engine of physical scene understanding'. In: *Proceedings of the National Academy of Sciences* 110.45 (2013), pp. 18327–18332. DOI: `10.1073/pnas.1306572110`.

[25] Tomer D. Ullman et al. 'Mind Games: Game Engines as an Architecture for Intuitive Physics'. In: *Trends in Cognitive Sciences* 21.9 (2017), pp. 649–665. DOI: `10.1016/j.tics.2017.05.012`.

[26] H Von Helmholtz. 'Physiological Optics'. In: *Uspekhi Fizicheskikh Nauk* III.10 (1925), pp. 1193–1213. DOI: `10.1007/978-3-540-39053-4`.