Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Behavioral Sciences

Predicate learning in neural systems: using oscillations to discover latent structure

Andrea E Martin[1] and Leonidas AA Doumas[2]

Humans learn to represent complex structures (e.g. natural language, music, mathematics) from experience with their environments. Often such structures are latent, hidden, or not encoded in statistics about sensory representations alone. Accounts of human cognition have long emphasized the importance of structured representations, yet the majority of contemporary neural networks do not learn structure from experience. Here, we describe one way that structured, functionally symbolic representations can be instantiated in an artificial neural network. Then, we describe how such latent structures (viz. predicates) can be learned from experience with unstructured data. Our approach exploits two principles from psychology and neuroscience: comparison of representations, and the naturally occurring dynamic properties of distributed computing across neuronal assemblies (viz. neural oscillations). We discuss how the ability to learn predicates from experience, to represent information compositionally, and to extrapolate knowledge to unseen data is core to understanding and modeling the most complex human behaviors (e.g. relational reasoning, analogy, language processing, game play).

Addresses
[1] Max Planck Institute for Psycholinguistics, The Netherlands
[2] University of Edinburgh, United Kingdom

Corresponding author: Martin, Andrea E (andrea.martin@mpi.nl)

Current Opinion in Behavioral Sciences 2019, **29**:77–83

This review comes from a themed issue on **Artificial intelligence**

Edited by **Matt Botvinick** and **Sam Gershman**

https://doi.org/10.1016/j.cobeha.2019.04.008

Introduction

As humans, we recognize our home, pet, or partner regardless of our viewing angle and the concomitant variation in the 2-D image on our retinas (e.g. Ref. [1]). Similarly, when we listen to speech or view sign, we understand linguistic structures that go far beyond any physical description of the stimulus (e.g. Refs. [2,3,4•]). Furthermore, we have the capacity to promiscuously apply what we know to new situations, for example, if we have to improvise a recipe with novel ingredients, we would never entertain cooking something by refrigerating it.[3] These examples emphasize several things. First, the ability to use 'incomplete' or partial sensory experience to infer the latent structures in the environment [5], and then reason and generalize based on these structures [6], appears to be crucial for everyday human behavior. Second, the domains where humans outperform artificial intelligence systems (AI) seem to involve inference beyond lower order statistical relationships [7]. While it is clear that, in the limit, AI can outmatch human performance on pure computation and statistical tasks (e.g. medical imaging), it is not clear how domains that require inference (e.g. analogy, scene comprehension), decision making (e.g. diagnosis, game play), or abstract rule generation (e.g. natural language) can be approached without a profound change in the principles of computation currently being espoused in the mainstream of both cognitive science and AI (for discussion see Refs. [4•,5–7]).

Here, we argue that *the capacity to learn structured (i.e. symbolic) representations from experience* underlies the flexible, extrapolatory nature of human behavior [6,8••,9]. We summarize the computational principles needed to instantiate structured representations (viz. predicates) in an artificial neural network [see also Ref. 10•], and we describe how predicates can be learned from unstructured data in an approach we call *predicate learning*. Predicate learning represents the integration of formal symbolic models with traditional neural computing principles and capitalizes on the information carried by oscillatory rhythms of neuronal computation.

The generalization problem and structured representations

Advances in AI and machine learning [11] have produced deep neural network (DNN) systems that reach and even exceed human levels of performance on a range of cognitive tasks [12]. DNNs can learn to perform a variety of tasks without any prior representations or knowledge (e.g. to play an Atari video game from pixel data and game scores, see Ref. [12]), but it is well-known that DNNs struggle with tasks that require generalization to input from outside the bounds of the training set (c.f. ranging from object recognition, inference, analogy, natural

[3] For recipes generated by deep learning networks, see https://www.dailydot.com/unclick/neural-network-recipe-generator/.

language; [7,8••]). DNNs' explicit (and intentional) lack of structured representations likely plays a role in this struggle, because accounts of how humans generalize tend to rely on powerful symbolic languages [6,10•,13]. An important reason these languages are so powerful is that they include predicates. A predicate is a data structure that can take (i.e. be bound to) arguments. Formally, a predicate is a function that takes some argument(s) and returns a truth value (e.g. specifying whether the argument(s) are members of a set). Functionally, a predicate can be understood as specifying a property about its arguments. For example, the predicate *red*(x) specifies the property of redness about the argument x.

Predicates are suitable means for the flexible transfer of information across contexts because the same representation can be used to effectively characterize wildly different input data (e.g. the predicate *contains* can be applied to broccoli and iron, but also to houses and rooms, or to first-order logic and quantification). However, the contemporary models that instantiate structured representations face a complementary challenge compared to DNNs: these structured models require specification, by the modeler, of a collection of necessary representational structures in advance of any actual learning; in other words, they do not learn the contents of their structures directly from the environment without the use of pre-specified representations and rules [cf. 9,14•,15,16]. That is, while structure-based models generalize more flexibly than DNNs, they do not perform general 'from scratch' learning because they feature symbolic representations that are specified a priori by the modeler [e.g. Refs. 9,15,16]. As a result, structured models often make strong nativist claims, for example, that a large set of representational elements and the rules for building compositions of these elements must be innate [17]. From a practical point of view, structured models that do no learn their structures can only be applied to problems for which a solution is already known because the relevant structures must be specified before the model runs.

Instantiation of predicates in artificial neural networks

A key notion for the instantiation of structured representations is *binding*. Importantly, the mechanism for binding predicates to arguments must meet two requirements [18,19••]. First, the mechanism that carries binding information must be completely independent of the representational elements that specify the identity of the active objects and predicates. For example, the representational elements *long-haired* and *cat*, and *short-haired* and *dog* might be bound to form the propositions *long-haired* (cat) and *short-haired*(dog). While the statement *long-haired*(cat) has meaning (a cat that has the property of having long hair) the elements *long-hair* and *cat* remain independent when so bound. That is, the predicate *long-hair* means the same thing whether it is bound to 'cat', 'dog', or 'automobile'. Second, the binding tag (the signal carrying the binding information) must be dynamic. That is, it must allow bindings to be created and destroyed on the fly. For instance, if the cat in the above example gets a short hair-cut, the binding of long-haired and cat must be broken, and the very-same representation of cat must be bound to the short-haired predicate to form *short-haired* (cat) where the same representational element coding for *short-haired* in *short-haired*(dog) is bound to exactly the same representational element coding for the cat in *long-haired*(cat).

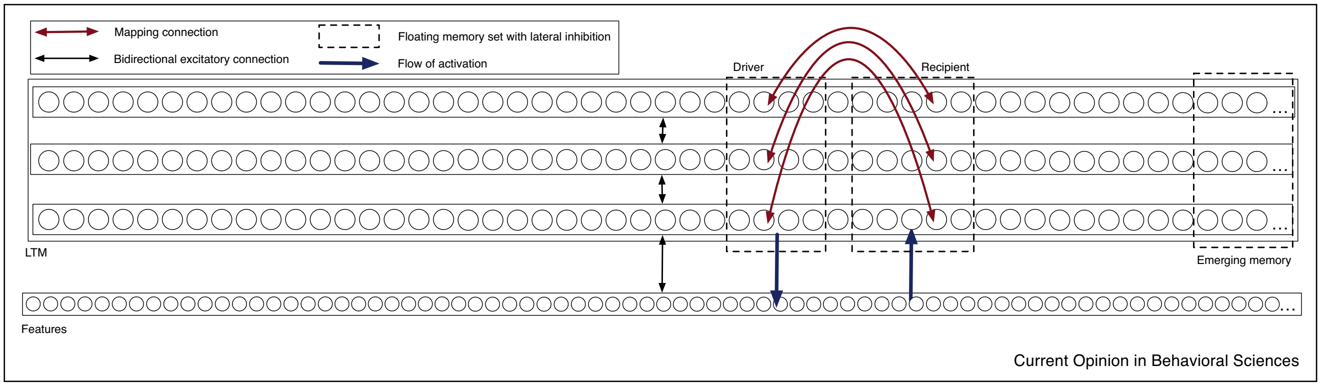
Binding of structured representations has been instantiated in neural networks in various forms since the early 1990s [19••,20–22]. The majority of approaches have used synchrony of firing to bind an argument [19••,20–22], though, we note synchrony-based systems do not learn predicates from unstructured data because they cannot separate predicates from their arguments without implementing separate data types a priori. Below we describe a predicate instantiation that exploits the asynchrony of unit firing (for the computing relevance of asynchrony see also Ref. [23]) in order to represent a predicate, role, and argument. The architecture, called DORA (*Discovery of Relations by Analogy*; [4•,8••,24•]), is descended from the symbolic-connectionist system LISA [*Learning and Inference with Schemas and Analogies*; 21,22]. DORA is based on two fundamental concepts from cognitive science and neuroscience: (1) that learning and generalization depend upon a process of comparison [25], and (2) that information in neural computing systems can be carried by the oscillations that emerge as its component units fire [19••,25,26•].

A model for predicate learning

DORA (*Discovery of Relations by Analogy*; Doumas *et al.* [24•]) is a neural network model that learns to represent structured (i.e. functionally symbolic) representations from unstructured examples without feedback. DORA is descended from of the symbolic-connectionist system LISA (*Learning and Inference with Schemas and Analogies*; [20,21]). DORA Below we describe DORA's architecture and operations only in functional terms for the purposes of brevity. The complete model including all implementational details can be found in Refs. [8••,24•].

The basic network macrostructure is presented in Figure 1. DORA consists of a long-term memory (LTM) composed of layers of bidirectionally connected units—we refer to these units as tokens. Token units are yoked to integrative inhibitors that integrate input from their yoked unit and active token units in higher layers, and fire after reaching a threshold. The yoked inhibitors serve the purpose of supporting phasic firing and implementing refractory periods in the token units. The bottom layer of token units is connected to a pool of feature units, which serve as distributed representations of objects in the world

Figure 1



Macrostructure of the DORA model. Adopted from Ref. [8••].

(initially), and (after learning) predicates. Features can be any kind of vector-based representations specified by the modeler, to raw pixels from an image.

Units in LTM become potentiated, and enter floating memory sets, which can be interpreted as analogues of attention and working memory (WM; [27,28]). One such set, the *driver*, corresponds to DORA's current focus of attention (e.g. a proposition in a story, or an image). A second set, the *recipient*, corresponds to DORA's active memory (AM; e.g. items from LTM that the DORA has retrieved based on its current focus of attention). A third set, the *emerging memory* (EM), corresponds to new or refined representations that the model learns (e.g. schemas; see Ref. [24•]). Token units within driver, recipient, and EM are laterally inhibitive (units in the same layer inhibit one another). The above is a way to interpret the function of these sets in the common jargon of cognitive psychology.

Activation in DORA flows from the driver to the recipient and the rest of LTM via the shared feature units. DORA's basic processing is summarized in Table 1. In brief, DORA starts with some representation in the driver. Activation flows from the driver to the rest of LTM via the shared feature units, and DORA will retrieve representations into AM (i.e. units from LTM become potentiated and enter AM; retrieval occurs via a Luce choice rule [29]). After retrieval, as units in the driver become active, they will produce patterns of activation on units in AM (again, via shared feature units). Excitatory connections, called mapping connections, are learned within-layer between co-active units in driver and recipient via a modified Hebbian algorithm [20,24•]. In accordance with any mapping connections DORA discovers, it will learn new representations or schemas, or it will perform relational generalization, the application of structure to another situation or set of inputs based on systematic correspondence between mapping connections.

Below we describe some of the key elements of DORA's processing in more detail. We focus on two key mechanisms, *time-based binding*, and new representation learning. We begin by describing the end state of DORA's learning: fully instantiated relational propositions. We then describe how DORA learns these representations from unstructured representations of objects. Full details of these operations are reported in [8••,24•].

Table 1

Basic processing in the model (adopted from Doumas *et al.* [24•], Appendix A)

1. Representations (objects or entire propositions) enter the driver.
2. Activation flows from the driver to the rest of the network via shared feature units.
3. If nothing in recipient:
a. DORA attempts retrieval via Luce choice rule.
4. If representations in recipient:
a. If no mapping connections:
i. Mapping via modified Hebbian algorithm (Hummel and Holyoak [14•]).
b. If mapping connections:
i. Learns new representations or refines representations via comparison-based learning (Doumas *et al.* [24•]).
ii. Generalizes via relational generalization algorithm (Doumas *et al.* [24•]; Hummel and Holyoak [21]).

Time-based binding

After learning, symbolic propositions are represented by a hierarchy of distributed and localist codes (see Figure 2a). At the lowest layer, feature units code the features of objects and roles in a distributed fashion. In the next layer, localist predicate–object units (POs) conjunctively code for individual predicates (or roles) and objects. In the next layer, localist role-binding units (RBs) link object and relational role PO units into specific role-filler pairs. Finally, localist P units link RBs into whole relational propositions. For example, a proposition like *contain* (obj1, obj2) is represented as the *container* role linked to obj1 via an RB unit, and the *contained* role to obj2 via an RB unit, and both of these RBs linked via a P unit to form the relational proposition *contain* (obj1, obj2).

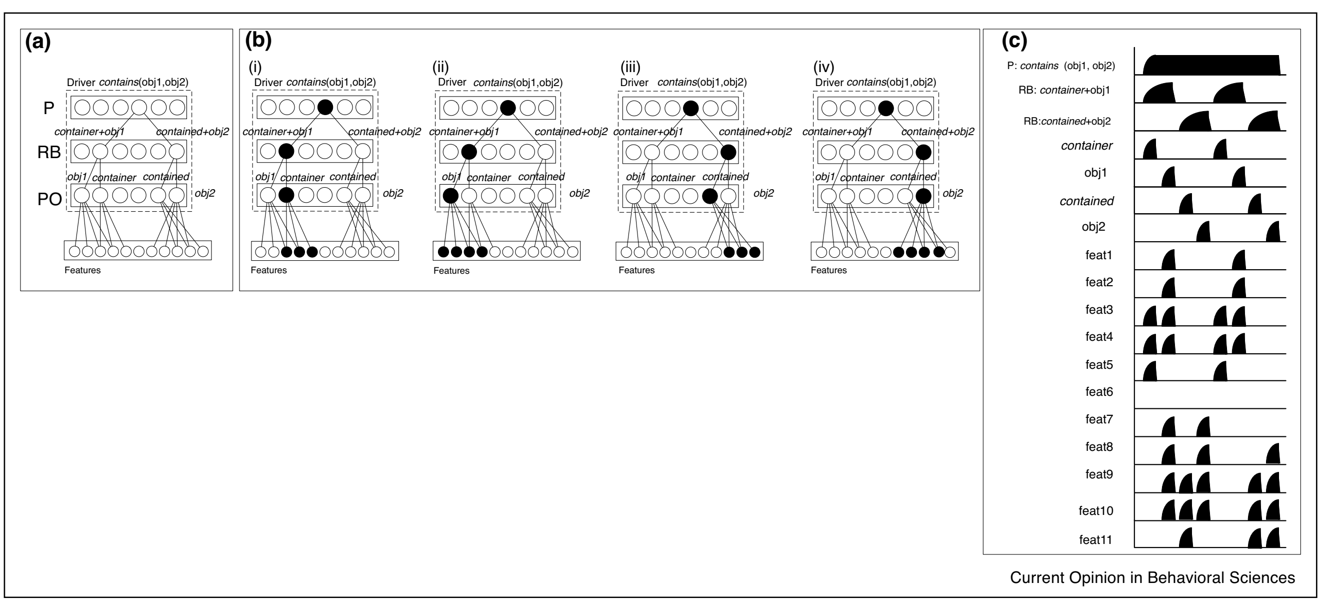
While this encoding is sufficient for long-term storage, it fails as an instantiation of dynamic binding: Binding information is carried by conjunctive units that definitionally defy predicate argument independence. In order to successfully instantiate functional predicates, the model must be able to dynamically bind predicates to arguments. In DORA, dynamic binding information is carried using time.

When a proposition like the one in Figure 2a is in the driver and becomes active, lateral inhibition and the yoked inhibitors will produce a systematic and repeating firing pattern. In brief, bound predicates and arguments will fire in direct sequence and out of synchrony with other bound predicates and arguments (Figure 2b). As the proposition becomes active (i.e. the P unit is activated), activation spreads to RB units which compete to become active. One of the RB units will win the competition, becoming more active and inhibiting the other (Figure 2bi). The active RB unit will activate its PO units, which will similarly compete to become active. The predicate might become active first (Figure 2bi), and after its yoked inhibitor fires, the bound argument will become active (Figure 2bii). When the active RB's yoked inhibitor fires, the next RB unit will become active (Figure 2biii) and will similarly activate its predicate and argument in sequence (Figure 2biii–iv). In short, binding information is carried dynamically in the units that maintain role-filler independence (the PO and feature units) by the sequence of firing (the same units could represent the inverse role-binding—*container* to obj2 and *contained* to obj1—simply by changing the order of firing). Figure 2c presents the same information in a wave diagram. These activation patterns give rise of oscillatory activity of units throughout the network, forming 'neural' oscillations.

Learning predicates using neural oscillations

At a basic level, DORA uses comparison to isolate shared properties of objects (represented in the feature unit layer) and to represent them as explicit structures. DORA starts with representations of objects encoded as simple

Figure 2



Symbolic propositions and binding in DORA. **(a)** A relational proposition represented in the DORA architecture. **(b)** Time-based binding (asynchronous or phase-lag 1) in DORA. (i) The representation, the token, and feature units representing the *container* role become active. (ii) The representation the token and feature units representing obj1 become active, marking it as bound to the *container* role (as they fire in direct sequence). (iii) The representation the token and feature units representing the *contained* role become active. (iv) The representation the token and feature units representing obj2 become active, marking it as bound to the *contained* role (as they fire in direct sequence). **(c)** Binding information from (b) represented in a wave diagram.

feature-vectors (i.e. a token unit connected to set of features describing that object). If DORA successfully maps an object in the driver to an object in the recipient, then these representations will become co-active, and corresponding features of the two representations will fire simultaneously, effectively comparing or superimposing the activation pattern of their features in the feature layer. For example, when DORA compares a square that is inside some object to a triangle that is inside some other object (e.g. the square inside the shield and triangle inside the circle in the first row of Figure 1), then the nodes representing the square and triangle fire together (Figure 3a). Any features that are shared by both compared objects (i.e. features common to both the square and the triangle) receive twice as much input and thus become roughly twice as active as features connected to one but not the other (Figure 3a). DORA uses a self-supervised learning algorithm we call comparison-based learning (CBL) to learn an explicit predicate representation of the featural overlap of the co-active objects.

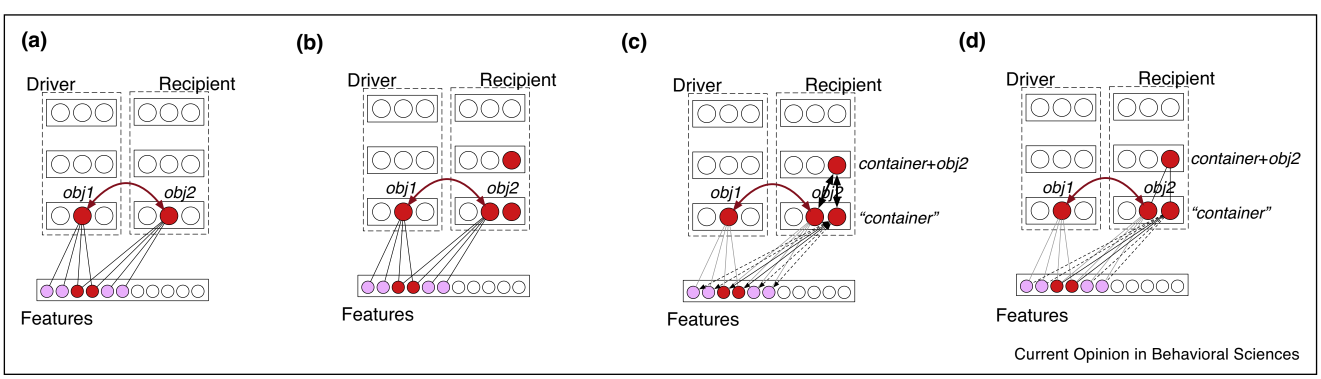
During CBL, for any layer above a layer with active tokens, DORA recruits and activates a token unit if none are already active (Figure 3b). When only single PO units are active, DORA also recruits and activates a PO token unit (Figure 3b). Connections between token units in adjacent layers are updated via a simple Hebbian rule. Because the strength of connections learned via Hebbian learning is a function of the units' activations, DORA learns stronger connections between the new PO unit and more active feature units (Figure 3c). The new PO thus becomes an explicit representation of the featural overlap of the compared objects (in this case the invariant properties of a 'container'; see Refs. [8••,23] for discussion of what these properties might be). In addition, DORA learns a conjunctive link between the recruited PO and the object in AM. The new PO unit serves as an explicit and functional single-place predicate (Figure 3d), dynamically bound to its object when if it enters the driver in the future (see above).

The same algorithm also allows DORA to link sets of co-occurring predicate-argument pairs into multi-place relations. If a set of predicate–object pairs co-occur they will be in the driver together. If DORA has previously encountered the same set of predicate–object pairs and encoded them in LTM, they can be retrieved into AM. When these representations are then mapped, CBL will result in a recruited P unit, which will learn connections to the RB units of the predicate–object pairs. The result is a multi-place symbolic relation similar to the one described in the previous section. Note that predicates and objects are not different datatypes in this architecture.

The DORA learning algorithm makes two interesting predictions about human mental representations. First, and most importantly, it suggests that we represent multi-place relations as linked sets of single-place predicates. Such a representational system is known as a role-binding calculus, and there exists a large body of evidence that human mental representations might indeed conform to it [18,20,30]. Second, it makes the prediction that humans should represent the constituent roles of a relation before they represent the relation as a unified whole. This prediction appears true of children (e.g. Ref. [31]).

DORA and the predicate learning approach account for a wide-range of phenomena in relational reasoning, analogy, cognitive development, and language processing (for a review, see Doumas and Martin [18]). Most recently, we have used the approach to demonstrate human level extrapolatory generalization in artificial environments [8••]. We augmented DORA with a simple visual pre-processor to perform object detection and allowed it to learn predicates from screen shots of the Atari game Breakout. We then used tabular q-learning to teach DORA to use the representations

Figure 3



Learning a new predicate representation in DORA. **(a)** Two objects (obj1 and obj2) are compared (i.e. co-activated) and mapped (solid red arrowed lines). **(b)** Units are recruited and activated in the RB layer and PO layer (see text). **(c)** DORA learned connections between active units via Hebbian learning (arrowed lines indicate newly learned connections; solid lines = stronger connections, dashed lines = weaker connections). **(d)** DORA has learned a representation of a new predicate that can be bound to obj2 via time-based binding.

that it had previously learned to play Breakout successfully. Breakout requires the player to move a paddle on the horizontally in order to hit a ball at bricks at the top of the screen. DORA was then able to transfer its knowledge of Breakout to the Atari game Pong, in which the player moves a paddle vertically to play a simple tennis-like game. Using the predicate representations that it had learned playing Breakout, DORA discovered the systematic correspondences between the two games (both involve keeping a ball in play using a paddle) and was able to successfully play Pong at above human levels with no additional training. By contrast, state of the art DNNs (e.g. a DQN based on Ref. [12]) completely failed to generalize to Pong based on training in Breakout. Our system was able to match and surpass human performance on Breakout and Pong, and importantly, it was also able to successfully return to playing Breakout after it played Pong, a simple task for humans that current non-structured systems fail at without specialized interleaved training routines.

Neural oscillations as the rhythms of computation

Predicate learning exploits a core set of neurophysiological computing principles, namely that computation in a neural network is rhythmic. Most crucially, predicates, once learned, are dynamically bound to their arguments by phase-lag, which is expressed as systematic asynchrony of unit firing [19••,23,26•], or desynchronization between the activation cycles of the nodes coding predicates and arguments (Figure 2). During asynchrony-based—or phase-lag-1—binding, as a predicate or proposition becomes active, bound arguments and predicates fire in direct sequence, and out of synchrony with other bound predicate-arguments sets. This feature is what allows the system to maintain independence between a predicate and its argument(s) and achieve variable-value independence [4•,10•,18]. At the same time, binding information is carried in the proximity of firing (e.g. with predicates firing directly before their arguments), meaning that representing predicates in a neural system relies critically on sensitivity to time, and rhythm, as dimensions of computation. Synchrony-based—or phase-lag-0—binding also occurs in the system depending on the computational goal, for example, a proposition can be activated by having its bound arguments and predicate fire together, but out of synchrony with other bound role-filler sets, in order to perform propositional-level computation of higher arities. By grouping representations into phase sets, or what is in and out of phase in the network, the system uses the rhythms of computation to both separate and combine information as needed.

Cortical oscillations have long been implicated as the indices of neural information processing [32]. Predicate learning in an artificial neural network relies on exploiting the naturally occurring 'neural' oscillations of distributed computation over time. Being sensitive to how information is carried in time in a neural system implies that the dynamics of the system can themselves be learned from. A similar principle appears in the dynamic reorganization of cortical networks during learning in humans (e.g. in Ref. [33]). Using oscillatory assembly activation to compute and to learn is potentially transformative, not only for its computational power (e.g. being able to learn from past states and learn relations over multiple time points and states), but also for the mechanistic link to neuroscientific theory and data (neural oscillations), and to formal accounts of cognition, including formalisms of natural language and predicate calculi [4•,6,15,16,18,34].Computing with neural oscillations represents a fundamental formal and neurophysiological synthesis between how human-like representations can be achieved in an artificial system that learns, and how distributed neural computing systems, including neuronal assemblies in biological brains, process information.

Predicate learning offers an account of how complex concepts might develop in neural computation systems without the need to hardwire or encode a priori structure, a theoretical and implementational limitation of current structure-based accounts of cognition (e.g. in Refs. [6,9,15,16]), and offers a solution to the classic generalization problem that unstructured deep-learning systems face (e.g. in Ref. [11,12]). A system that uses predicate learning can discover, and predicate, what is latent in the environment, and discover what is relevant for behavior. Predicate learning ultimately relies on the capacity of a system to be compositional — to host representations that can be combined without changing core representations in order to flexibly generate new representations as the environment and behavior require.

In sum, we have described in brief how predicates can be learned from unstructured data using rhythmic, desynchronized neural oscillations. Learning symbolic structure from signals that naturally occur in distributed computing systems offers a promising approach whereby the computational principles that can yield the highest forms of the human mind (e.g. relational reasoning, formal and natural language processing) can also be realized in systems based on the computational primitives of neurophysiology.

Conflict of interest statement

Nothing declared.

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Olshausen BA: **Perception as an inference problem**. In *The Cognitive Neurosciences V*. Edited by Gazzaniga M, Mangun R. MIT Press; 2014.

2. Ding N, Melloni L, Zhang H, Tian X, Poeppel D: **Cortical tracking of hierarchical linguistic structures in connected speech**. *Nat Neurosci* 2016, **19**:158.

3. Martin AE: **Language processing as cue integration: Grounding the psychology of language in perception and neurophysiology**. *Front Psychol* 2016, **7**.

4. • Martin AE, Doumas LA: **A mechanism for the cortical computation of hierarchical linguistic structure**. *PLoS Biol* 2017, **15**:e2000663.
A first demonstration that a symbolic predicate-calculus-based neural network model expends energy and oscillates like human cortical networks do when processing the same sentence structures.

5. Gershman SJ, Niv Y: **Learning latent structure: carving nature at its joints**. *Curr Opin Neurobiol* 2010, **20**:251-256.

6. Kemp C, Tenenbaum JB: **Structured statistical models of inductive reasoning**. *Psychol Rev* 2009, **116**:20.

7. Bowers JS: **Parallel distributed processing theory in the age of deep networks**. *Trends Cogn Sci* 2017, **21**:950-961.

8. •• Doumas LA, Puebla G, Martin AE: **Human-like generalization in a machine through predicate learning**. *arXiv preprint arXiv* 2018. 1806.01709.
The first demonstration, to our knowledge, of a neural network showing one-shot generalization to data from outside bounds of the training set. A predicate learning system successfully plays the horizontally oriented Atari game Pong after being trained only on the vertically oriented Atari game Breakout, and vice versa. A concise list procedure of the computation-algorithm-mapping for each step of predicate learning is available in the Supplemental materials.

9. Lake BM, Salakhutdinov R, Tenenbaum JB: **Human-level concept learning through probabilistic program induction**. *Science* 2015, **350**:1332-1338.

10. • Hummel JE: **Getting symbols out of a neural architecture**. *Connect Sci* 2011, **23**:109-118.
A clear description of how to solve C. R. Gallistel's problem (as discussed in the excellent *Memory and the Computational Brain*, Wiley-Blackwell) of representing symbols with neural computing principles (not an account of how symbols are learned in neural systems).

11. LeCun Y, Bengio Y, Hinton G: **Deep learning**. *Nature* 2015, **521**:436-444.

12. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G *et al.*: **Human-level control through deep reinforcement learning**. *Nature* 2015, **518**:529.

13. Anderson JR: *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press; 2007.

14. • Hummel JE, Holyoak KJ: **Distributed representations of structure: a theory of analogical access and mapping**. *Psychol Rev* 1997, **104**:427.
The classic first instance of a symbol system in a distributed neural network that can solve analogies.

15. Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND: **How to grow a mind: statistics, structure, and abstraction**. *Science* 2011, **331**:1279-1285.

16. Kemp C: **Exploring the conceptual universe**. *Psychol Rev* 2012, **119**:685.

17. Carey S: *The Origin of Concepts*. Oxford University Press; 2009.

18. Doumas LA, Martin AE: **Learning structured representations from experience**. *Psychol Learn Motiv* 2018, **69**:165-203.

19. •• von der Malsburg C: **Binding in models of perception and brain function**. *Curr Opin Neurobiol* 1995, **5**:520-526.
An early and elegant espousal of the power of including using time and neural synchrony to perform binding.

20. Hummel JE, Holyoak KJ: **Distributed representations of structure: a theory of analogical access and mapping**. *Psychol Rev* 1997, **104**:427.

21. Hummel JE, Holyoak KJ: **A symbolic-connectionist theory of relational inference and generalization**. *Psychol Rev* 2003, **110**:220.

22. Shastri L: **Advances in Shruti—a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony**. *Appl Intell* 1999, **11**:79-108.

23. Love Bradley C: **Utilizing time: asynchronous binding**. *Adv Neural Info Process Syst* 1999.

24. • Doumas LAA, Hummel JE, Sandhofer CM: **A theory of the discovery and predication of relational concepts**. *Psychol Rev* 2008, **115**:1-43.
The full model description of the architectures, algorithms, and principles needed to learn relational predicates from flat feature vectors. Also features simulations of human data from the literature on developmental relational reasoning.

25. Holyoak KJ, Thagard P: *Mental Leaps: Analogy in Creative Thought*. MIT Press; 1996.

26. • von der Malsburg C: **Am I thinking assemblies?** *Brain Theory*. Berlin, Heidelberg: Springer; 1986, 161-176.
A thoughtful, pithy consideration of what it would mean for neuronal assemblies to be human thinking.

27. Cowan N: **The magical number 4 in short-term memory: a reconsideration of mental storage capacity**. *Behav Brain Sci* 2001, **24**:87-114.

28. McElree B: **Accessing recent events**. *Psychol Learn Motiv* 2006, **46**:155-200.

29. Luce RD: **On the possible psychophysical laws**. *Psychol Rev* 1959, **66**:81.

30. Livins KA, Doumas LA, Spivey MJ: **Shaping relations: exploiting relational features for visuospatial priming**. *J Exp Psychol: Learn Mem Cogn* 2016, **42**:127.

31. Smith LB, Rattermann MJ, Sera M: **"Higher" and "lower": comparative and categorical interpretations by children**. *Cogn Dev* 1988, **3**:341-357.

32. Buzsáki G: *Rhythms of the Brain*. Oxford University Press; 2006.

33. Bassett DS, Wymbs NF, Porter MA, Mucha PJ, Carlson JM, Grafton ST: **Dynamic reconfiguration of human brain networks during learning**. *Proc Natl Acad Sci U S A* 2011, **108**:7641-7646.

34. Partee BB, ter Meulen AG, Wall R: *Mathematical Methods in Linguistics*. Springer Science & Business Media; 2012.