The Logic of Hebbian Learning

Caleb Kisby and **Saúl A. Blanco**
Department of Computer Science, Indiana University, Bloomington, IN 47408, USA
{cckisby, sblancor}@indiana.edu

Lawrence S. Moss
Department of Mathematics, Indiana University, Bloomington, IN 47405, USA
lmoss@indiana.edu

Abstract

We present the logic of Hebbian learning, a dynamic logic whose semantics[1] are expressed in terms of a layered neural network learning via Hebb's associative learning rule. Its language consists of modality $\mathbf{T}\varphi$ (read "typically φ," formalized as forward propagation), conditionals $\varphi \Rightarrow \psi$ (read "typically φ are ψ"), as well as dynamic modalities $[\varphi^+]\psi$ (read "evaluate ψ after performing Hebbian update on φ"). We give axioms and inference rules that are sound with respect to the neural semantics; these axioms characterize Hebbian learning and its interaction with propagation. The upshot is that this logic describes a neuro-symbolic agent that both learns from experience and also reasons about what it has learned.

Introduction

Artificial intelligence has long been marked by a schism between two of its major paradigms: symbolic reasoning and connectionist learning. Neural systems have had wild success with learning from unstructured data, whereas symbolic reasoners are notorious for their rigidity. On the other hand, symbolic systems excel at sophisticated (static) reasoning tasks that neural systems cannot readily learn. Symbolic systems also tend to have more explainable reasoning, thanks to their use of explicit inferences in an intuitive language. Moreover, due to their connection with logic, it is straightforward to compare the relative power and complexity of different symbolic reasoners.

But as Valiant famously put it, intelligent cognitive agents must have *both* "the ability to learn from experience, and the ability to reason from what has been learned" (Valiant 2003). *Neuro-symbolic artificial intelligence* has emerged in the last few decades to address this challenge — a monumental effort to integrate neural and symbolic systems, while retaining the advantages of both (see (Bader and Hitzler 2005) and (Sarker et al. 2021), two surveys that span the decades). Despite the cornucopia of neuro-symbolic proposals, the field has not yet agreed on an interface between the two that satisfyingly preserves both flexible learning and expressive reasoning.

Following the path set out by (Balkenius and Gärdenfors 1991) and (Leitgeb 2001; 2003), we advance the following proposal for the neuro-symbolic interface. Rather than viewing the neural and symbolic as two different systems to be combined, we view them as two ways of interpreting the same agent. More precisely, we view the dynamics of neural networks as the semantics to a formal logic. This logic serves as a bridge between the neural network model and formal inference.

Previous work, particularly (Leitgeb 2001), has considered how forward propagation in binary feed-forward nets forms a sound and complete semantics for the (static) conditional logic **CL** (*loop-cumulative*). The novelty of our paper is that we extend this logic by viewing a simple learning policy — Hebbian update ("neurons that fire together wire together") — as a dynamic modality. By doing so, we demonstrate that the dynamics of Hebbian learning (in binary feed-forward nets) directly corresponds to a particular dynamic multimodal logic that we call *the logic of Hebbian learning*. This logic meets Valiant's challenge: It characterizes a cognitive agent that can learn from experience and also reason about what it has learned.

Our main result is the soundness of axioms and inference rules that characterize Hebbian learning. The most interesting axioms involve the interaction between Hebbian update and forward propagation. We also demonstrate how our logic models the learning of a concrete neural network. And although we leave the question of completeness open, we close by considering the importance of completeness for logics of this kind.

Related Work

Logics with Neural Semantics. The idea that we can view neural networks as the semantics for symbolic reasoning dates back to (McCulloch and Pitts 1943). Our work builds on a recent reimagining of this à la (Balkenius and Gärdenfors 1991), (Leitgeb 2001; 2003; 2018), which formally characterize the dynamics of inhibitory neural networks as conditional logics. Similarly, (Blutner 2004) demonstrates that Hopfield networks correspond to the logic of what he calls "weight-annotated Poole systems." More recently, (Giordano, Gliozzi, and Dupré 2021) describe multilayer perceptrons and self-organizing maps in terms of typicality in defeasible description logics. Yet no neural semantics to date has tackled

[1] A Python implementation of our semantics, using Tensorflow & Keras (Abadi and others 2015)), is available at https://github.com/ais-climber/neural-semantics

the issue of learning — doing this for Hebbian learning is precisely the contribution of our paper.

Neuro-Symbolic AI. Across the neuro-symbolic literature, an ubiquitous premise is that integration involves combining or composing otherwise distinct neural and symbolic modules. In contrast, this paper presents the neural and symbolic as two perspectives we can have about the same agent.

To our knowledge, the combined work of (Garcez, Broda, and Gabbay 2001) and (Garcez, Lamb, and Gabbay 2008) is the only neuro-symbolic proposal (besides neural semantics, see above) that exhibits this intimate interface between the two. The former gives a formally sound method for extracting conditionals from a network and the latter gives a method for build neural network models from rules (in a variety of different logics). When combined, we can freely translate between a neural network and its beliefs. But unlike our work, this framework does not offer a logical account of the neural network's learning.

Dynamic Logics for Learning. Two recent papers, (Baltag et al. 2019) and (Baltag, Li, and Pedersen 2019), also present dynamic multimodal logics that characterize learning. The former models an individual's learning in the limit, whereas the latter models supervised learning as a game played between student and teacher. But it is unclear how learning policies expressed in these logics might relate to specific neural implementations of learning such as Hebbian update and backpropagation.

Furthermore, the syntax and inferences of our logic do not resemble either of these in a meaningful way. Perhaps the closest logics to ours are dynamic logics of *preference upgrade*, in the sense of (Van Benthem and Liu 2007). In particular, consider the modalities $[\Uparrow\varphi]$ (lexicographic upgrade) and $[\uparrow\varphi]$ (elite change) (Van Benthem 2007). Both of these operators implement policies for modifying an agent's preference relation $<$ over possible worlds. As with our logic, the key axioms characterizing these policies deal with their interaction with conditionals $\varphi \Rightarrow \psi$. But the semantics of our logic have a different flavor; we leave the issue of how our neural semantics relate to classical preference relations to future work. In addition, both $[\Uparrow\varphi]$ and $[\uparrow\varphi]$ are reducible to the static language of conditionals, whereas it is presently unclear how our $[\varphi^+]$ might reduce to its base language.

Background

Neural Network Models

A model of the logic of Hebbian learning is just a special type of artificial neural network that we call a *binary feedforward neural network* (BFNN).

Definition 1. A BFNN is a pointed directed graph $\mathcal{N} = \langle N, E, W, A, O, \eta \rangle$, where

- N is a finite nonempty set (the set of neurons)
- $E \subseteq N \times N$ (the set of excitatory connections)
- $W : N \times N \to \mathbb{R}$ (the weight of a given connection)
- A is a function which maps each $n \in N$ to $A^{(n)} : \mathbb{R}^k \to \mathbb{R}$ (the activation function for n, where k is the indegree of n)
- O is a function which maps each $n \in N$ to $O^{(n)} : \mathbb{R} \to \{0, 1\}$ (the output function for n)
- $\eta \in \mathbb{R}, \eta \geq 0$ (the learning rate)

Moreover, BFNNs are *feed-forward*, i.e. they do not contain cycles of edges with all nonzero weights. BFNNs are also *binary*, i.e. the output of each neuron is in $\{0, 1\}$. This binary assumption is unrealistic in practice, although letting it go is just a matter of extending our two-valued logic towards a fuzzy-valued logic (left to future work).

We further require that each composition of activation and output functions $O^{(n)} \circ A^{(n)}$ is *strictly* monotonically increasing, i.e. for all $\vec{x}, \vec{y} \in \mathbb{R}^k$,

$$\text{If } \vec{x} < \vec{y} \text{ then } O^{(n)}(A^{(n)}(\vec{x})) < O^{(n)}(A^{(n)}(\vec{y}))$$

Our activation functions include in particular those sigmoid functions commonly used for neural networks in practice.

We write W_{ij} to mean $W(i, j)$, for $(i, j) \in E$. When m_i is drawn from a sequence $m_1, \ldots, m_k$, we write $A^{(n)}(\overrightarrow{W}(m_i, n))$ as shorthand instead of the full expression $A^{(n)}(W(m_1, n), \ldots, W(m_k, n))$.

The Dynamics of Propagation

Of course, BFNNs are not merely static directed graphs, but are dynamic in nature. When a BFNN receives a signal (which we model as the initial state), it propagates that signal forward until the state of the net stabilizes. This stable state of the net is considered to contain the net's response (answer) to the given signal (question). We model forward propagation as follows, drawing heavily from the approach proposed by (Leitgeb 2001).[2]

We consider a neuron n active if its activation $A^{(n)}$ triggers an output $O^{(n)}$ of 1 (intuitively, if the neuron fires). Since our BFNNs are binary, either a given neuron is active (1) or it is not (0). So we can identify the state of $\mathcal{N}$ with the set of neurons that are active. For a given BFNN $\mathcal{N}$, let its set of states be

$$\mathsf{Set} = \{S \mid S \subseteq N\}$$

Neurons in a state $S \in \mathsf{Set}$ can subsequently activate new neurons, which activate yet more neurons, until eventually the state of $\mathcal{N}$ stablizes. We call this final state of affairs $\mathsf{Prop}(S)$, the *propagation* of S.

Definition 2. Let $\mathsf{Prop} : \mathsf{Set} \to \mathsf{Set}$ be defined recursively as follows: $n \in \mathsf{Prop}(S)$ iff either

(Base Case) $n \in S$, or

(Constructor) For those $m_1, \ldots, m_k \in \mathsf{Prop}(S)$ such that $(m_i, n) \in E$ we have

$$O^{(n)}(A^{(n)}(\overrightarrow{W}(m_i, n))) = 1$$

[2] Note that (Leitgeb 2001) defines propagation for *inhibition nets*, i.e. weightless BFNNs with both excitatory and inhibitory connections. But this paper also proves that inhibition nets and BFNNs are equivalent with respect to their propagation structure. So we import the ideas and results here.

Alternatively, consider a finite automaton with state space Set and transition function $F_{S^*} : \mathsf{Set} \to \mathsf{Set}$ tracking the propagation of an initial state S^* through $\mathcal{N}$. We can view $\mathsf{Prop}(S^*)$ as a fixed point of F_{S^*} (Leitgeb 2001).

The key insight of (Leitgeb 2001) is that we can neatly characterize the algebraic structure of Prop as a closure operator.

Theorem 1. Let $\mathcal{N} \in \mathsf{Net}$. For all $S, S_1, S_2 \in \mathsf{Set}$,

- **(Inclusion)** $S \subseteq \mathsf{Prop}(S)$
- **(Idempotence)** $\mathsf{Prop}(S) = \mathsf{Prop}(\mathsf{Prop}(S))$
- **(Cumulative)** If $S_1 \subseteq S_2 \subseteq \mathsf{Prop}(S_1)$ then $\mathsf{Prop}(S_1) = \mathsf{Prop}(S_2)$
- **(Loop)** If $S_1 \subseteq \mathsf{Prop}(S_0), \ldots, S_n \subseteq \mathsf{Prop}(S_{n-1})$ and $S_0 \subseteq \mathsf{Prop}(S_n)$, then $\mathsf{Prop}(S_i) = \mathsf{Prop}(S_j)$ for all $i, j \in \{0, \ldots, n\}$

Crucially, Prop is not monotonic — it is not the case that for all $S_1, S_2 \in \mathsf{Set}$, if $S_1 \subseteq S_2$, then $\mathsf{Prop}(S_1) \subseteq \mathsf{Prop}(S_2)$. The reason for this is that a BFNN's weights W_{ij} can be negative, which allows S_2 to inhibit the activation of new neurons that were otherwise activated by S_1. But in place of monotonicity, Prop is *loop-cumulative* (in the terminology of (Kraus, Lehmann, and Magidor 1990)).

From Hebbian Learning to Logic

The Dynamics of Hebbian Learning

The plan from here is to extend this logic of propagation by providing an account of Hebbian learning. Our goal is to cast Hebbian update as a dynamic modality, so that we can explore its interactions with Prop in symbolic language. As with Prop, we start by outlining the algebraic structure of Hebbian update.

Hebb's classic learning rule (Hebb 1949) states that when two adjacent neurons are simultaneously and persistently active, the connection between them strengthens. In contrast with, e.g. backpropagation, Hebbian learning is errorless and unsupervised. Another key difference is that Hebbian update is local — the change in a weight ΔW_{ij} depends only on the activation of the immediately adjacent neurons. For this reason, the Hebbian family of learning policies has traditionally been considered more biologically plausible than backpropagation. There are many variations of Hebbian learning, but we only consider the most basic (unstable, no weight decay) form of Hebb's rule: $\Delta W_{ij} = \eta x_i x_j$, where η is the learning rate and x_i, x_j are the outputs of adjacent neurons i and j, respectively.

In order to incorporate Hebb's rule into our framework, we introduce a function Inc ("increase the weights") to strengthen those edges in a BFNN $\mathcal{N}$ whose neurons are active when we feed $\mathcal{N}$ a signal $S \in \mathsf{Set}$.

Definition 3. For $S \in \mathsf{Set}$, let $\chi_S : N \to \{0, 1\}$ be given by $\chi_S(n) = 1$ iff $n \in S$

Definition 4. Let $\mathsf{Inc} : \mathsf{Net} \times \mathsf{Set} \to \mathsf{Net}$ be given by $\mathsf{Inc}(\langle N, E, W, A, O, \eta\rangle, S) = \langle N, E, W^*, A, O, \eta\rangle$, where

$$W^*_{ij} = W_{ij} + \eta \cdot \chi_{\mathsf{Prop}(S)}(i) \cdot \chi_{\mathsf{Prop}(S)}(j)$$

Notice that we propagate S before getting the active status of neurons. This is because otherwise we would never strengthen connections beyond the input layer.

We were able to formulate the algebraic properties of Prop in terms of Set containment. Similarly, we express the properties of Inc in terms of Net containment.

Definition 5. Let $\mathcal{N}_1, \mathcal{N}_2 \in \mathsf{Net}$ differ only in their weights. We write

$$\mathcal{N}_1 \preceq \mathcal{N}_2$$

to mean that for all $S \in \mathsf{Set}$, $\mathsf{Prop}_{\mathcal{N}_1}(S) \subseteq \mathsf{Prop}_{\mathcal{N}_2}(S)$. We use $\cong$ to express that $\mathcal{N}_1 \preceq \mathcal{N}_2$ and $\mathcal{N}_2 \preceq \mathcal{N}_1$.

For example, $\mathsf{Inc}(\mathcal{N}, S)$ is a supernet of $\mathcal{N}$ because strengthening weights via the Inc operation only has the potential to *expand* future propagations. To further cement this intuition, consider the least upper bound $\mathcal{N}^{lub}$ of $\preceq$. $\mathcal{N}^{lub}$ is that net whose weights have been "maximally" strengthened, and so every propagation $\mathsf{Prop}(S)$ results in the entire set N.

We have the following test to determine if $\mathcal{N}_1 \preceq \mathcal{N}_2$.

Lemma 2. Suppose $\mathcal{N}_1$ and $\mathcal{N}_2$ are the same except for their weights, and let $S \in \mathsf{Set}$. Then $\mathsf{Prop}_{\mathcal{N}_1}(S) \subseteq \mathsf{Prop}_{\mathcal{N}_2}(S)$ iff for all $n \in \mathsf{Prop}_{\mathcal{N}_1}(S)$ and for those $m_1, \ldots, m_k \in \mathsf{Prop}_{\mathcal{N}_1}(S)$ such that $(m_i, n) \in E$,

$$\begin{array}{c} O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}_1}(m_i, n))) = 1 \\ \text{implies} \\ O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}_2}(m_i, n))) = 1 \end{array} \qquad (*)$$

Corollary 1. Let $\mathcal{N}_1, \mathcal{N}_2$ be the same except for their weights. Then $\mathcal{N}_1 \preceq \mathcal{N}_2$ iff for all $n \in N$ and for those $m_1, \ldots, m_k \in N$ such that $(m_i, n) \in E$, $(*)$ holds.

Notice that the right-hand side of this biconditional does not mention Prop; this test reduces the *dynamic* condition ($\mathcal{N}_1 \preceq \mathcal{N}_2$) to a *static* condition (a statement about the relative weights between $\mathcal{N}_1$ and $\mathcal{N}_2$).

These two facts are exactly what we need for the following algebraic characterization of Inc.

Theorem 3. For all $\mathcal{N}, \mathcal{N}_1, \mathcal{N}_2 \in \mathsf{Net}$ and $S, S_1, S_2 \in \mathsf{Set}$, Inc satisfies

- **(Inclusion)** $\mathcal{N} \preceq \mathsf{Inc}(\mathcal{N}, S)$
- **(Absorption)** $\mathsf{Inc}(\mathcal{N}, \mathsf{Prop}(S)) \cong \mathsf{Inc}(\mathcal{N}, S)$
- **(Monotonicity in $\mathcal{N}$)** if $\mathcal{N}_1 \preceq \mathcal{N}_2$ then $\mathsf{Inc}(\mathcal{N}_1, S) \preceq \mathsf{Inc}(\mathcal{N}_2, S)$
- **(Local)** $$\mathsf{Prop}_{\mathsf{Inc}(\mathcal{N}, S_2)}(S_1) \subseteq \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$$
- **(Cumulative)** If $\mathsf{Prop}_{\mathcal{N}}(S_1) \subseteq \mathsf{Prop}_{\mathcal{N}}(S_2)$ and $\mathsf{Prop}_{\mathcal{N}}(S_2) \subseteq \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_1)$, then $\mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_1) = \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_2)$
- **(Loop)** If $\mathsf{Prop}_{\mathcal{N}}(S_1) \subseteq \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_0)$, $\ldots, \mathsf{Prop}_{\mathcal{N}}(S_n) \subseteq \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_{n-1})$, and $\mathsf{Prop}_{\mathcal{N}}(S_0) \subseteq \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_n)$, then $\mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_i) = \mathsf{Prop}_{\mathsf{Inc}(\mathcal{N},S)}(S_j)$ for all $i, j \in \{0, \ldots, n\}$

Like Prop, Inc is loop-cumulative (in S). But also like Prop, Inc is not monotonic in S. For a counterexample of S-monotonicity, see Figure 2 (discussed in more detail towards the end of this paper).

Syntax and Semantics

We can now introduce the logic of Hebbian learning. Let $p, q, \ldots$ be finitely many propositional variables. These represent fixed, 'ontic' states, i.e. established choices of neurons that correspond to features in the external world. For example, p might be the set of neurons that encapsulates the color *pink*. We presume that we already agree on these states, although we acknowledge that this is a major unresolved empirical issue. As for more complex formulas:

Definition 6. Formulas of our language $\mathcal{L}$ are given by

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \varphi \Rightarrow \varphi \mid \mathbf{T}\varphi \mid [\varphi^{+}]\varphi$$

where p is any propositional variable. We define $\bot$, $\vee$, $\leftrightarrow$, $\Leftrightarrow$, and their duals $\langle\mathbf{T}\rangle$, $\langle\varphi^{+}\rangle$ in the usual way.

The modalities $\mathbf{T}$ and $[\varphi^{+}]$ reflect our two operations Prop and Inc, respectively. We intend for $\mathbf{T}\varphi$ to denote "the propagation of signal φ," and for $[\varphi^{+}]\psi$ to denote "after performing Hebbian update on φ, evaluate ψ." We import $\varphi \Rightarrow \psi$ from (Leitgeb 2001), read "the propagation of signal φ contains ψ". Note that $\varphi \Rightarrow \psi$ is redundant (equivalent to $\mathbf{T}\varphi \to \psi$ using the semantics below), though we keep it in our syntax because it conveniently expresses "the net *classifies* φ as ψ" (if φ is interpreted as an input and ψ as a classification).

Our formulas also have more classical alternative readings, divorced from the dynamics of neural networks. Following (Leitgeb 2001), we will define $\varphi \Rightarrow \psi$ such that it has the conditional reading "typically φ are ψ" (where φ and ψ are read as generics, e.g. "typically birds fly"). This gives us a natural preferential reading for $\mathbf{T}\varphi$ as "typically φ" or "the typical φ."[3] Finally, Hebbian learning $[\varphi^{+}]\psi$ has a dual reading as *preference upgrade* (Van Benthem and Liu 2007). As mentioned in the Related Work section, we leave the question concerning how $[\varphi^{+}]$ can be viewed classically as updating a preference relation to future work.

A model of our logic is just a BFNN $\mathcal{N}$ equipped with an interpretation function $[\![\cdot]\!] : \mathcal{L} \to \mathsf{Set}_{\mathcal{N}}$.

Definition 7. Let $\mathcal{N} \in \mathsf{Net}$. Our semantics are defined recursively as follows:

$$\boxed{\begin{array}{lcl}
[\![p]\!] & & \in \mathsf{Set} \text{ is fixed, nonempty} \\
[\![\top]\!] & = & \emptyset \\
[\![\neg\varphi]\!] & = & \overline{[\![\varphi]\!]} \\
[\![\varphi \wedge \psi]\!] & = & [\![\varphi]\!] \cup [\![\psi]\!] \\
[\![\varphi \to \psi]\!] & = & [\![\top]\!] \text{ iff } [\![\varphi]\!] \supseteq [\![\psi]\!], \text{ else } [\![\bot]\!] \\
[\![\varphi \Rightarrow \psi]\!] & = & [\![\top]\!] \text{ iff } \mathsf{Prop}([\![\varphi]\!]) \supseteq [\![\psi]\!], \text{ else } [\![\bot]\!] \\
[\![\mathbf{T}\varphi]\!] & = & \mathsf{Prop}([\![\varphi]\!]) \\
[\![[\varphi^{+}]\psi]\!] & = & [\![\psi]\!]_{\mathsf{Inc}(\mathcal{N}, [\![\varphi]\!])}
\end{array}}$$

It may seem odd that we interpret $\wedge$ as union (instead of intersection), $\to$ as superset (instead of subset), and $\top$ as $\emptyset$ (instead of N). But this choice reflects the intuition that neurons act as "elementary-feature-detectors" (Leitgeb 2001). For example, say $[\![\varphi]\!]$ represents those neurons that are *necessary* for detecting an apple, and $[\![\psi]\!]$ represents those neurons that are *necessary* for detecting the color red. If the net observes a red apple ($\varphi \wedge \psi$), both the neurons detecting red-features $[\![\varphi]\!]$ and the neurons detecting apple-features $[\![\psi]\!]$ necessarily activate, i.e. $[\![\varphi]\!] \cup [\![\psi]\!]$ activates. As for implication, "every apple is red" ($\varphi \to \psi$) holds for a net iff whenever the neurons detecting apple-features $[\![\varphi]\!]$ necessarily activate, so do the neurons detecting red-features $[\![\psi]\!]$. But this is only true if $[\![\varphi]\!] \supseteq [\![\psi]\!]$. This justifies us reading propositional connectives classically, despite the backwards flavor of the semantics.

Our interpretation of formulas is completely algebraic, in the sense that formulas denote sets rather than truth-values. But we can consider formulas to have truth-values as follows.

Definition 8. $\mathcal{N} \models \varphi$ iff $[\![\varphi]\!]_{\mathcal{N}} = \emptyset$.

This choice also appears to be strange at its surface. But it is a natural one in light of the fact that we defined $[\![\top]\!] := \emptyset$. For example, consider implication: $\mathcal{N} \models \varphi \to \psi$ holds iff $[\![\varphi \to \psi]\!] = \emptyset = [\![\top]\!]$, which holds iff $[\![\varphi]\!] \supseteq [\![\psi]\!]$ by our semantics.

A curious consequence is that if $\mathcal{N} \models \varphi$ and φ cannot be written to contain an implication $\to$, then φ must be a tautology. But we do not consider this troubling, since it only makes sense to consider a neural network's judgment of φ when given a state $[\![\psi]\!]$ the net is in.

Inference and Axioms

The proof system for our logic is as follows. We have $\vdash \varphi$ iff either φ is an axiom, or φ follows from previously obtained formulas by one of the inference rules. If $\Gamma \subseteq \mathcal{L}$ is a set of formulas, we consider $\Gamma \vdash \varphi$ to hold whenever there exist finitely many $\psi_1, \ldots, \psi_k \in \Gamma$ such that $\vdash \psi_1 \wedge \ldots \wedge \psi_k \to \varphi$.

We list the axioms and inference rules for our logic in Figure 1. Our main result is the soundness of these axioms and rules — we do not claim that this list forms a complete axiomatization (we revisit the question of completeness in the Conclusion).

The static base of our logic can either be viewed as the conditional logic **CL** (loop-cumulative), or alternatively as the modal logic characterized by $\mathbf{T}$ along with inference rules ($\mathrm{C}_{\Rightarrow}$), ($\textsc{Loop}_{\Rightarrow}$) expressing the loop-cumulativity of Prop. Linking these perspectives is the rule ($\textsc{Typ}$). As a modality, $\mathbf{T}$ is neither normal, regular, nor monotonic, but it is classical. Note for instance that the normal modal property (K) (expressed in terms of $\mathbf{T}$) is equivalent to

$$(\mathrm{K}) \quad \mathbf{T}(\varphi \wedge \psi) \leftrightarrow (\mathbf{T}\varphi \wedge \mathbf{T}\psi)$$

neither direction of which is sound in our logic.

As with $\mathbf{T}$, we have the inference rules ($\textsc{Nec}_{+}$), (C_{+}), ($\textsc{Loop}_{+}$) for $[\varphi^{+}]$ (the latter two express the loop-cumulativity of Inc). Since Hebbian update only affects the propagation of states, we have reduction axioms (R_p), ($\mathrm{R}_{\neg}$), ($\mathrm{R}_{\wedge}$), as well as a reduction axiom ($\textsc{Nest}_{\mathbf{T}}$) for terms that nest $\mathbf{T}$ within $[\varphi^{+}]$.

[3] Our notation takes inspiration from (Giordano, Gliozzi, and Dupré 2021), which formalizes the dynamics of a net via a concept constructor $\mathbf{T}$ in the description logic $\mathcal{ALC}$. Note the subtle difference between their typicality inclusions $\mathbf{T}(\varphi) \sqsubseteq \psi$ and our $\mathbf{T}\varphi \to \psi$: Ours flips the direction of containment.

	Basic Axioms
(PC)	All propositional tautologies
(Dual)	$\langle \mathbf{T} \rangle \varphi \leftrightarrow \neg \mathbf{T} \neg \varphi$
(N)	$\mathbf{T}\top$
(T)	$\mathbf{T}\varphi \to \varphi$
(4)	$\mathbf{T}\varphi \to \mathbf{T}\mathbf{T}\varphi$
	Inference Rules
(MP)	$\dfrac{\varphi \quad \varphi \to \psi}{\psi}$
(Typ)	$\dfrac{\varphi \Rightarrow \psi}{\mathbf{T}\varphi \to \psi} \quad \dfrac{\mathbf{T}\varphi \to \psi}{\varphi \Rightarrow \psi}$
($\mathrm{C}_{\Rightarrow}$)	$\dfrac{\varphi \to \psi \quad \psi \Rightarrow \varphi}{\varphi \Leftrightarrow \psi}$
($\mathrm{Loop}_{\Rightarrow}$)	$\dfrac{\varphi_0 \Rightarrow \varphi_1 \cdots \varphi_{k-1} \Rightarrow \varphi_k \quad \varphi_k \Rightarrow \varphi_0}{\varphi_0 \Rightarrow \varphi_k}$
(Nec_{+})	$\dfrac{\psi}{[\varphi^+]\psi}$
(C_{+})	$\dfrac{\psi \to \rho \quad [\varphi^+]\rho \to \psi}{[\varphi^+]\psi \leftrightarrow [\varphi^+]\rho}$
(Loop_{+})	$\dfrac{[\varphi^+]\psi_0 \to \psi_1 \cdots [\varphi^+]\psi_{k-1} \to \psi_k \quad [\varphi^+]\psi_k \to \psi_0}{[\varphi^+]\psi_0 \to \psi_k}$
	Reduction Axioms
(R_p)	$[\varphi^+]p \leftrightarrow p$
($\mathrm{R}_\neg$)	$[\varphi^+]\neg\psi \leftrightarrow \neg[\varphi^+]\psi$
($\mathrm{R}_\wedge$)	$[\varphi^+](\psi \wedge \rho) \leftrightarrow ([\varphi^+]\psi \wedge [\varphi^+]\rho)$
($\mathrm{Nest}_{\mathbf{T}}$)	$[\mathbf{T}\varphi^+]\psi \leftrightarrow [\varphi^+]\psi$
	Key Axioms
(NS)	$[\varphi^+]\mathbf{T}\psi \to \mathbf{T}[\varphi^+]\psi$
(TP)	$\mathbf{T}[\varphi^+]\psi \wedge \mathbf{T}\varphi \to [\varphi^+]\mathbf{T}\psi$

Figure 1: A list of sound rules and axioms of the logic of Hebbian learning. We leave the question of completeness to future work.

In lieu of a full reduction for $[\varphi^+]\mathbf{T}\psi$, we instead have the weaker axioms (NS) and (TP). These two axioms capture key cognitive biases of a Hebbian agent. Consider the axiom (TP), i.e. (Typicality Preservation)

$$\text{(TP)} \quad \mathbf{T}[\varphi^+]\psi \wedge \mathbf{T}\varphi \to [\varphi^+]\mathbf{T}\psi$$

This says that if our agent expects ψ is normally true after learning φ, but she also happens to expect φ, then after learning φ the typicality of ψ will be preserved. This is a peculiar kind of cognitive bias whereby a Hebbian agent maintains her prior attitudes when presented with news she already expects.

The axiom (NS), i.e. (No Surprises)

$$\text{(NS)} \quad [\varphi^+]\mathbf{T}\psi \to \mathbf{T}[\varphi^+]\psi$$

says that if after learning φ, our agent thinks normally ψ, then she would have expected ψ to be true after learning φ in the first place. Loosely: She will never be surprised.

Soundness of these axioms is just a matter of matching each axiom with its corresponding property of Inc.

Theorem 4. The rules and axioms above are sound, i.e. hold for all $\mathcal{N} \in \mathsf{Net}$.

Applying the Logic: A Concrete Example

We now demonstrate our neuro-symbolic interface by way of an example neural network in a machine learning context. The task: Given an image of an animal, classify it as flying or non-flying. Suppose we have the partially pre-trained BFNN $\mathcal{N}$ in Figure 2.

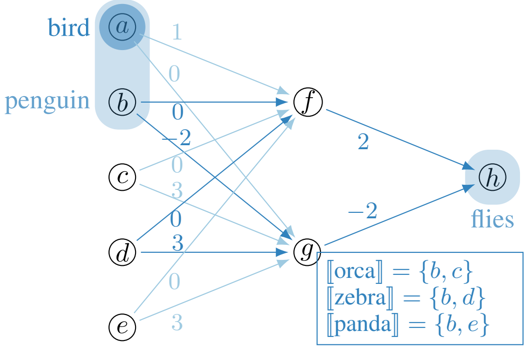
bird
penguin
a
b
c
d
e
f
g
h
flies
1
0
0
−2
0
3
0
3
0
3
2
−2
$[\![\text{orca}]\!] = \{b, c\}$
$[\![\text{zebra}]\!] = \{b, d\}$
$[\![\text{panda}]\!] = \{b, e\}$

Figure 2: A BFNN $\mathcal{N}$, equipped with the ReLU activation function, $T = 1$, and $\eta = 1$. After observing the dataset $\langle \text{orca}, \text{zebra}, \text{panda} \rangle$, $\mathcal{N}$ learns that penguins do not fly, while preserving the fact that birds typically fly.

For simplification's sake, let's suppose that our animal images can be reduced to 5-dimensional vectors in order to be fed into the input layer of $\mathcal{N}$. Say:

penguin	$\langle 11000 \rangle$	orca	$\langle 01100 \rangle$
zebra	$\langle 01010 \rangle$	panda	$\langle 01001 \rangle$

In addition, suppose an image activates the first node if and only if it depicts a bird.

We can identify each animal with the set of nodes it activates in the input layer. This gives us the sets shown in Figure 2. We can also identify the class of things that fly with the output node, i.e. $[\![\text{flies}]\!] = \{h\}$. In principle we can identify propositions with sets containing hidden nodes as well, although in practice the meaning of hidden nodes is often unclear.

With this interpretation in mind, we see that $\mathcal{N} \models \text{bird} \Rightarrow \text{flies}$, but also $\mathcal{N} \models \text{penguin} \Rightarrow \text{flies}$ (which is incorrect). Our hope is that $\mathcal{N}$ corrects this mistake via Hebbian learning.

Say we expose $\mathcal{N}$ to non-flying animals that share the black-and-white color of penguins, e.g. we train $\mathcal{N}$ on the dataset $\langle \text{orca}, \text{zebra}, \text{panda} \rangle$. The propagations of each instance will increase W_{bg}. Once we have given $\mathcal{N}$ the entire dataset ($W_{bg} = 1$), $\mathsf{Prop}([\![\text{penguin}]\!])$ will contain g, which will cancel the signal given by $f \to h$. Our logic successfully models this behavior:

$$\mathcal{N} \models [\text{orca}^+][\text{zebra}^+][\text{panda}^+](\text{bird} \Rightarrow \text{flies}), \text{ yet}$$
$$\mathcal{N} \not\models [\text{orca}^+][\text{zebra}^+][\text{panda}^+](\text{penguin} \Rightarrow \text{flies})$$

i.e. $\mathcal{N}$ learns that penguins do not fly while preserving the fact that birds typically fly.

As it happens, if we modify $\mathcal{N}$ such that $W_{bg} = 0$ then this serves as a counterexample to monotonicity in S (see the discussion following Theorem 3). In particular, we have $\mathcal{N} \models \mathbf{T}(\text{penguin}) \to \text{flies}$, yet $\mathcal{N} \not\models [\text{orca}^+]\mathbf{T}(\text{penguin}) \to [\text{orca}^+]\text{flies}$.

Conclusion and Future Work

In this paper, we gave sound axioms and rules characterizing the logic of Hebbian learning. This logic interfaces the neuro-symbolic divide by characterizing conditionals $\Rightarrow$ and modalities $\mathbf{T}$, $[\varphi^+]$ in terms of the propagation and Hebbian update of signals in a neural network. The upshot of all this is that this logic describes a neuro-symbolic agent that learns associatively and also reasons about what it has learned.

We leave open the question of whether the axioms and rules we list are complete. But we take this opportunity to stress the importance of having strong completeness for logics of this kind. Strong completeness for a *static* neural semantics provides a bridge across which we can extract a set of rules Γ from an interpreted network, and also build an interpreted neural network implementing Γ. But once the neural network updates, we lose the interpretations of neurons that allow for these translations. If we had strong completeness for the *dynamic* logic, we could fully track the interpretations while the net learns and preserve this neuro-symbolic correspondence.

Beyond the logic of Hebbian learning, we believe that this framework will be a fruitful way to explore the neuro-symbolic interface for a variety of neural networks and learning policies. Exciting future directions include:

1. Mapping more expressive syntax to neural activity
2. Generalizing to a broader class of neural networks
3. Generalizing to a broader class of activation functions
4. Characterizing other learning policies in logical terms

The holy grail of this line of work is to completely axiomatize the (1) first-order logic of (2) nonbinary (fuzzy-valued) neural networks with (3) more varied (e.g. ReLU and GELU) activation functions that (4) learn via backpropagation.

Acknowledgements

We thank the anonymous reviewers for their careful reviews and helpful comments. Additionally, C. Kisby was supported in part by the US Department of Defense [Contract No. W52P1J2093009].

References

Abadi, M., et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Bader, S., and Hitzler, P. 2005. Dimensions of neural-symbolic integration-a structured survey. *arXiv preprint cs/0511042*.

Balkenius, C., and Gärdenfors, P. 1991. Nonmonotonic Inferences in Neural Networks. In *KR*, 32–39.

Baltag, A.; Gierasimczuk, N.; Özgün, A.; Sandoval, A. L. V.; and Smets, S. 2019. A dynamic logic for learning theory. *Journal of Logical and Algebraic Methods in Programming* 109:100485.

Baltag, A.; Li, D.; and Pedersen, M. Y. 2019. On the right path: a modal logic for supervised learning. In *International Workshop on Logic, Rationality and Interaction*, 1–14. Springer.

Blutner, R. 2004. Nonmonotonic inferences and neural networks. In *Information, Interaction and Agency*. Springer. 203–234.

Garcez, A. d.; Broda, K.; and Gabbay, D. M. 2001. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence* 125(1-2):155–207.

Garcez, A. S.; Lamb, L. C.; and Gabbay, D. M. 2008. *Neural-symbolic cognitive reasoning*. Springer Science & Business Media.

Giordano, L.; Gliozzi, V.; and Dupré, D. T. 2021. From common sense reasoning to neural network models through multiple preferences: An overview. *CoRR* abs/2107.04870.

Hebb, D. 1949. *The organization of behavior: A neuropsychological theory*. New York: Wiley.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44(1-2):167–207.

Leitgeb, H. 2001. Nonmonotonic reasoning by inhibition nets. *Artificial Intelligence* 128(1-2):161–201.

Leitgeb, H. 2003. Nonmonotonic reasoning by inhibition nets II. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 11(supp02):105–135.

Leitgeb, H. 2018. Neural Network Models of Conditionals. In *Introduction to Formal Philosophy*. Springer. 147–176.

McCulloch, W. S., and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5(4):115–133.

Sarker, M. K.; Zhou, L.; Eberhart, A.; and Hitzler, P. 2021. Neuro-symbolic artificial intelligence: Current trends. *arXiv preprint arXiv:2105.05330*.

Valiant, L. G. 2003. Three problems in computer science. *Journal of the ACM (JACM)* 50(1):96–99.

Van Benthem, J., and Liu, F. 2007. Dynamic logic of preference upgrade. *Journal of Applied Non-Classical Logics* 17(2):157–182.

Van Benthem, J. 2007. Dynamic logic for belief revision. *Journal of applied non-classical logics* 17(2):129–155.