THE LOGIC OF HEBBIAN LEARNING

Caleb Kisby[1], **Saúl A. Blanco**[1], **Lawrence S. Moss**[2]
[1]Department of Computer Science, Indiana University
[2]Department of Mathematics, Indiana University
Bloomington, IN 47408, USA
{cckisby, sblancor, lmoss}@indiana.edu

Note: This arXiv print is a fuller version of our FLAIRS paper [Kisby et al., 2022], complete with proofs we could not include in the conference format. We have also corrected some errors, as well as certain aesthetic choices that made the proofs and logic somewhat awkward. The original paper is available here.

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

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

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 et al., 2021] describe multilayer perceptrons and self-organizing maps in terms of defeasible description logics. Yet no neural semantics to date has tackled 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 et al., 2001] and [Garcez et al., 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., 2019a] and [Baltag et al., 2019b], 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 are very different; 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)
- $A : \mathbb{Q} \to \mathbb{Q}$ (the activation function)
- $\eta \in \mathbb{R}, \eta \geq 0$ (the learning rate)

Moreover, BFNNs are *feed-forward*, i.e. they do not contain cycles of edges with all nonzero weights. This allows us to partition a BFNN's nodes into layers as follows. For every neuron $n \in N$, $\mathsf{layer}(n)$ is the maximal length of a path from any node m to n, where m has no predecessors.

BFNNs are also *binary*, i.e. $A : \mathbb{Q} \to \{0, 1\}$ is a binary step function. This binary assumption is clearly unrealistic in practice. Letting it go is a matter of extending our two-valued logic towards a fuzzy-valued logic, which we leave to future work.

We further require that the activation function A is strictly increasing, i.e. for all $\vec{x}, \vec{y} \in \mathbb{R}^k$ if $\vec{x} < \vec{y}$ then $A(\vec{x}) < A(\vec{y})$. We will more often refer to the equivalent condition: $\vec{x} \leq \vec{y}$ iff $A(\vec{x}) \leq A(\vec{y})$. Note that this increasing assumption holds for sigmoid activation functions commonly used for neural networks in practice.

We write W_{mn} to mean $W(m, n)$, for $(m, n) \in E$. We also write $\deg(n)$ to indicate the degree (number of predecessors) of 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].

We consider a neuron n active if its activation $A^{(n)}$ is high enough to trigger 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

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

We can get the activation value of a particular neuron n in a state S using the following characteristic function.

Definition 2. For $S \in \mathsf{State}$, let $\chi_S : N \to \{0, 1\}$ be given by $\chi_S(n) = 1$ iff $n \in S$

Neurons in a state $S \in \mathsf{State}$ 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 3. Let $\mathcal{N}$ be a BFNN, $n \in N$, and let $\vec{m} = m_1, \ldots, m_{\deg(n)}$ list the predecessors of n. We define $\mathsf{Prop}_{\mathcal{N}} : \mathsf{State} \to \mathsf{State}$ recursively on $l = \mathsf{layer}(n)$ as follows.

Base Case $(l = 0)$**.** $n \in \mathsf{Prop}_{\mathcal{N}}(S)$ iff $n \in S$
Constructor $(l \geq 0)$**.** $n \in \mathsf{Prop}_{\mathcal{N}}(S)$ iff either $n \in S$ or n is activated by its predecessors $m_i \in \mathsf{Prop}_{\mathcal{N}}(S)$, i.e.

$$A(\sum_{i=1}^{\deg(n)} W_{m_i n} \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m_i)) = 1$$

Alternatively, consider a finite automaton with state space State and transition function $F_{S^*} : \mathsf{State} \to \mathsf{State}$ 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 $S, S_1, S_2 \in \mathsf{State}$, Prop satisfies

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 $S_1 \subseteq \mathsf{Prop}(S_0), \ldots, S_k \subseteq \mathsf{Prop}(S_{k-1})$ and $S_0 \subseteq \mathsf{Prop}(S_k)$,
then $\mathsf{Prop}(S_i) = \mathsf{Prop}(S_j)$ for all $i, j \in \{0, \ldots, k\}$

Proof. For the full proof, we refer the reader to Leitgeb [2001]. Note that this paper actually defines Prop for *inhibition nets*, i.e. weightless BFNNs with both excitatory and inhibitory connections. But the paper later shows that inhibition nets and BFNNs are equivalent with respect to their propagation structure, and so we can import the results here.

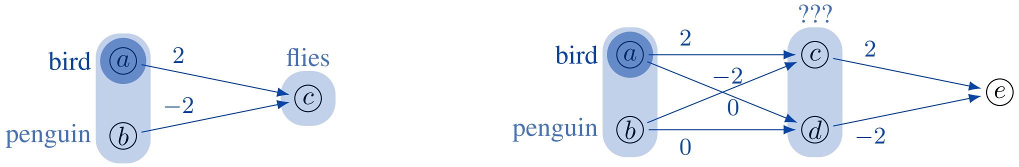

Figure 1: (Left) The BFNN $\mathcal{N}_1$ illustrates how BFNNs leverage negative weights to perform nonmonotonic inferences. (Right) TODO

Regardless, we will walk through the proof of the (Loop) property using our weighted BFNNs to help the reader get a feel for how these arguments typically go.

Let $k \geq 0$ and suppose the hypothesis. Our goal is to show that for each i, $\mathsf{Prop}(S_i) \subseteq \mathsf{Prop}(S_{i-1})$, and additionally $\mathsf{Prop}(S_0) \subseteq \mathsf{Prop}(S_k)$. This will show that all $\mathsf{Prop}(S_i)$ contain each other, and so are equal. Let $i \in \{0, \ldots, k\}$ (if $i = 0$ then $i - 1$ refers to k), and let $n \in \mathsf{Prop}(S_i)$. We proceed by induction on $\mathsf{layer}(n)$.

Base Step. At layer 0, $\mathsf{Prop}(S_i) = S_i$. And so $n \in S_i$. But since $S_i \subseteq \mathsf{Prop}(S_{i-1})$, we have $n \in \mathsf{Prop}(S_{i-1})$.

Inductive Step. Let $\mathsf{layer}(n) \geq 0$. Since $n \in \mathsf{Prop}(S_i)$, we have two cases: Either $n \in S_i$ or n is activated by its predecessors. The first case is similar to the base step. As for the second case, for those $\vec{m} = m_1, \ldots, m_k$ such that $(m_h, n) \in E$ we have

$$A\Big(\sum_{i=1}^{\deg(n)} W_{m_h n} \cdot \chi_{\mathsf{Prop}(S_i)}(m_h)\Big) = 1$$

Our inductive hypothesis says that for all predecessors m_h, $m_h \in \mathsf{Prop}(S_i) \leftrightarrow m_h \in \mathsf{Prop}(S_j)$. In particular, this is true for S_i and S_{i-1}. So we can substitute $\chi_{\mathsf{Prop}(S_{i-1})}(m_h)$ for $\chi_{\mathsf{Prop}(S_i)}(m_h)$ in the inner expression, which immediately gives us $n \in \mathsf{Prop}(S_{i-1})$ by definition. □

Crucially, Prop is not monotonic; it is not the case that for all $S_1, S_2 \in \mathsf{State}$, if $S_1 \subseteq S_2$, then $\mathsf{Prop}(S_1) \subseteq \mathsf{Prop}(S_2)$. This is because the net's weights can be negative, and so $\mathsf{Prop}(S_2)$ can inhibit the activation of new neurons that would otherwise be activated by $\mathsf{Prop}(S_1)$. In place of monotonicity, Prop is *loop-cumulative* (in the terminology of [Kraus et al., 1990]).

The BFNN $\mathcal{N}_1$ in Figure 1 illustrates how BFNNs encode nonmonotonic conditionals using negative weights. $\mathcal{N}_1$ has three basic states: 'bird' $= \{a\}$, 'penguin' $= \{a, b\}$, and 'flies' $= \{c\}$. Each state contains the set of neurons that detect a feature of the input. Note that 'penguin' *contains* 'bird'. This models the fact that that all penguins are birds; equivalently, all bird-features are penguin-features.

For concreteness, say our activation function is a step function with a threshold of 0. Observe that when the input is a bird, the resulting activated state $\mathsf{Prop}(\text{bird}) = \{a, c\}$ contains 'flies'. But when the input is a penguin, the negative weight inhibits the 'flies' node from activating. In this way, $\mathcal{N}_1$ classifies birds as flying, yet also successfully classifies penguins as non-flying.

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 Hebb ("perform one step of Hebbian update") to strengthen those edges in a BFNN $\mathcal{N}$ whose neurons are active when we feed $\mathcal{N}$ a signal $S \in \mathsf{State}$.

Definition 4. Let $\mathsf{Hebb} : \mathsf{Net} \times \mathsf{State} \to \mathsf{Net}$ be given by $\mathsf{Hebb}(\langle N, E, W, A, O, \eta \rangle, S) = \langle N, E, W^*, A, O, \eta \rangle$, where for all neurons $i, j \in N$,

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

Notice that we strengthen edges within $\mathsf{Prop}(S)$ rather than just S. 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 State containment. Similarly, we express the properties of Hebb 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 $S \in \mathsf{State}$, $\mathsf{Prop}_{\mathcal{N}_1}(S) \subseteq \mathsf{Prop}_{\mathcal{N}_2}(S)$. We use $\cong$ to express that $\mathcal{N}_1 \preceq \mathcal{N}_2$ and $\mathcal{N}_2 \preceq \mathcal{N}_1$.

For example, $\mathsf{Hebb}(\mathcal{N}, S)$ is a supernet of $\mathcal{N}$ because strengthening weights via the Hebb 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 can now prove a sound algebraic characterization of Hebb.

Theorem 2. For all $\mathcal{N}, \mathcal{N}_1, \mathcal{N}_2 \in \mathsf{Net}$ and $S, S_1, S_2 \in \mathsf{State}$, Hebb satisfies

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

Proof. TODO

Inclusion. Suppose $n \in \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$. By induction on $\mathsf{layer}(n)$:

Base Step. At layer 0, $\mathsf{Prop}_{\mathcal{N}}(S_2) = S_2$. But then by inclusion, $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N}, S_2)}(S_1)$.

Inductive Step. Let $\mathsf{layer}(n) \geq 0$. Since $n \in \mathsf{Prop}_{\mathcal{N}}(S_2)$, we have two cases: Either $n \in S_2$ or n is activated by its predecessors. The first case is similar to the base step. As for the second case, for those $\vec{m} = m_1, \ldots, m_k$ such that $(m_i, n) \in E$ we have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}(\mathcal{N}, S_2)}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N}, S_2)}(S_1)}(m_i)) = 1$$

Absorption.

$$\begin{array}{rll} & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathsf{Hebb}(\mathcal{N}, \mathsf{Prop}_{\mathcal{N}}(S))}(m_i, n))) & \\ = & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}}(m_i, n) + \eta \overrightarrow{\chi}_{\mathsf{Prop}(\mathsf{Prop}(S))}(m_i, n))) & \text{(by Definition 4)} \\ = & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}}(m_i, n) + \eta \overrightarrow{\chi}_{\mathsf{Prop}(S)}(m_i, n))) & \text{(by Idempotence)} \\ = & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathsf{Hebb}(\mathcal{N}, S)}(m_i, n))) & \text{(by Definition 4)} \end{array}$$

Since $O^{(n)}$ is binary, (**??**) and its converse hold.

Monotonicity in $\mathcal{N}$.

$$\begin{array}{rll} & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathsf{Hebb}(\mathcal{N}_1, S)}(m_i, n))) & \\ \leq & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}_1}(m_i, n) + \eta \overrightarrow{\chi}_{\mathsf{Prop}_{\mathcal{N}_1}(S)}(m_i, n))) & \text{(by Definition 4)} \\ \leq & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathcal{N}_2}(m_i, n) + \eta \overrightarrow{\chi}_{\mathsf{Prop}_{\mathcal{N}_2}(S)}(m_i, n))) & (\dagger) \\ \leq & O^{(n)}(A^{(n)}(\overrightarrow{W}_{\mathsf{Hebb}(\mathcal{N}_2, S)}(m_i, n))) & \text{(by Definition 4)} \end{array}$$

which again shows that (**??**) holds. To see $(\dagger)$, notice that from $\mathcal{N}_1 \preceq \mathcal{N}_2$ and (**??**) we have that each

$$O^{(n)}(A^{(n)}(W_{\mathcal{N}_1}(m_i, n))) \leq O^{(n)}(A^{(n)}(W_{\mathcal{N}_2}(m_i, n)))$$

The backwards direction of (**??**) gives

$$W_{\mathcal{N}_1}(m_i, n) \leq W_{\mathcal{N}_2}(m_i, n)$$

In addition, $\mathcal{N}_1 \preceq \mathcal{N}_2$ by definition means that for all S, $\mathsf{Prop}_{\mathcal{N}_1}(S) \subseteq \mathsf{Prop}_{\mathcal{N}_2}(S)$. This in turn implies that each

$$\chi_{\mathsf{Prop}_{\mathcal{N}_1}(S)}(m_i, n) \leq \chi_{\mathsf{Prop}_{\mathcal{N}_2}(S)}(m_i, n)$$

And so the inner sum for $\mathcal{N}_1$ is component-wise $\leq$ the inner sum for $\mathcal{N}_2$. Since $O^{(n)} \circ A^{(n)}$ is increasing (**??**), we get (†).

Local. Suppose $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)$. By induction on $\mathsf{layer}(n)$:

Base Step. At layer 0, $\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1) = S_1$. But then by inclusion, $n \in \mathsf{Prop}_{\mathcal{N}}(S_1)$. And so $n \in \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$.

Inductive Step. Let $\mathsf{layer}(n) \geq 0$. Since $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)$, we have two cases: Either $n \in S_1$ or n is activated by its predecessors. The first case is similar to the base step. As for the second case, for those $\vec{m} = m_1, \ldots, m_k$ such that $(m_i, n) \in E$ we have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}(\mathcal{N},S_2)}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i)) = 1$$

Expanding out the $W_{\mathsf{Hebb}(\mathcal{N},S_2)}$, we get

$$A(\sum_{i=1}^{\deg(n)} (W_{\mathcal{N}}(m_i, n) + \eta \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(n)) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i)) = 1$$

Now suppose for contradiction that $n \notin \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$. In particular, $n \notin \mathsf{Prop}_{\mathcal{N}}(S_2)$, and so $\chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i) = \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(n) = 0$. So we have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i)) = 1$$

By our inductive hypothesis, for all m_i, if $m_i \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)$ then $m_i \in \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$. So

$$\begin{array}{rcl} \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i) & \leq & \chi_{\mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i) \\ & = & \max(\chi_{\mathsf{Prop}_{\mathcal{N}}(S_1)}(m_i), \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i)) \end{array}$$

Note that either $\chi_{\mathsf{Prop}_{\mathcal{N}}(S_1)}(m_i) \leq \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i)$ or $\chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i) < \chi_{\mathsf{Prop}_{\mathcal{N}}(S_1)}(m_i)$. Consider the first case (the second is similar). The above inequality simplifies to

$$\chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i) \leq \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i)$$

ERROR!!! The following has a mistake — the sum of zero terms isn't necessarily zero!!!
Additionally — and crucially — $\chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i) > 0$; since otherwise, we would have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i, n) \cdot 0) = A(0) = 1$$

which would contradict the fact that A is zero at zero. Since this term is positive, we can apply the multiplication property for inequalities to get

$$W_{\mathcal{N}}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i) \leq W_{\mathcal{N}}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i)$$

At last, applying the fact that A is nondecreasing,

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S_2)}(S_1)}(m_i)) \leq A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i, n) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S_2)}(m_i))$$

So this latter term is equal to 1, which by definition of Prop means that $n \in \mathsf{Prop}_{\mathcal{N}}(S_2)$. But this contradicts our assumption that $n \notin \mathsf{Prop}_{\mathcal{N}}(S_1) \cup \mathsf{Prop}_{\mathcal{N}}(S_2)$!

Cumulative & Loop. Finally, the Cumulative and Loop properties follow from the same properties for Prop in Theorem 1 by substituting, for each $i \in \{0, \ldots, n\}$,

$$\begin{array}{rcl} \mathcal{N} & \rightsquigarrow & \mathsf{Hebb}(\mathcal{N}, S) \\ S_i & \rightsquigarrow & \mathsf{Prop}_{\mathcal{N}}(S_i) \end{array}$$

□

Proposition 1. Hebb is not monotonic in S.

Proof. **FILL IN** □

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

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

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

The modalities $\mathbf{T}$ and $[\varphi^+]$ reflect our two operations Prop and Hebb, 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."[2] 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 $[\![\cdot]\!] : \mathcal{L} \to \mathsf{State}_{\mathcal{N}}$.

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

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

Notice that these semantics are "flipped" in the sense that $\wedge$ is interpreted as union (instead of intersection), and consequently $\to$ is interpreted as superset (instead of subset). This choice may seem odd, but it 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 $[\![\psi]\!]$.

[2] Our notation takes inspiration from [Giordano et al., 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.

But this is only true if $\llbracket \varphi \rrbracket \supseteq \llbracket \psi \rrbracket$. 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 $\llbracket \varphi \rrbracket_{\mathcal{N}} = \emptyset$.

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

Inference and Axioms

Basic Axioms and Inference Rules

(PC)	All propositional tautologies
(MP)	$\frac{\varphi \quad \varphi \to \psi}{\psi}$
(NEC$_{\mathbf{T}}$)	$\frac{\varphi}{\mathbf{T}\varphi}$
(NEC$_{+}$)	$\frac{\psi}{[\varphi^+]\psi}$

Reduction Axioms

(R$_p$)	$[\varphi^+]p \leftrightarrow p$
(R$_\neg$)	$[\varphi^+]\neg\psi \leftrightarrow \neg[\varphi^+]\psi$
(R$_\wedge$)	$[\varphi^+](\psi \wedge \rho) \leftrightarrow ([\varphi^+]\psi \wedge [\varphi^+]\rho)$

T Axioms

(LOOP)	$(\mathbf{T}\varphi_0 \to \varphi_1) \wedge \cdots \wedge (\mathbf{T}\varphi_k \to \varphi_0)$ $\to (\mathbf{T}\varphi_0 \leftrightarrow \mathbf{T}\varphi_k)$
(DUAL)	$\langle \mathbf{T} \rangle \varphi \leftrightarrow \neg \mathbf{T} \neg \varphi$
(T)	$\mathbf{T}\varphi \to \varphi$
(4)	$\mathbf{T}\varphi \to \mathbf{TT}\varphi$

Interaction Axioms

(NEST$_{\mathbf{T}}$)	$[\mathbf{T}\varphi^+]\psi \leftrightarrow [\varphi^+]\psi$
(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 2: A list of sound rules and axioms of the logic of Hebbian learning. We leave the question of completeness to future work.

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 2. 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 is the modal logic characterized by $\mathbf{T}$. If we translate $\varphi \Rightarrow \psi$ via $\mathbf{T}\varphi \to \psi$ as before, we see that this modal logic contains the conditional logic **CL** (loop-cumulative). 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

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

neither direction of which is sound in our logic. Instead, we have the (LOOP) axiom expressing the loop-cumulativity of Prop. In fact, (LOOP) is all that is needed for this weakening of monotonicity — note that we have dropped cumulativity (C$_\Rightarrow$), as well as (LOOP$_+$) and (C$_+$) for $[\varphi^+]$, since they all follow from the present axioms (compare against [Kisby et al., 2022]).

Since Hebbian update only affects the propagation of states, we have reduction axioms (R$_p$), (R$_\neg$), (R$_\wedge$), as well as the axiom (NEST$_{\mathbf{T}}$) for terms that nest $\mathbf{T}$ within $[\varphi^+]$.

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.

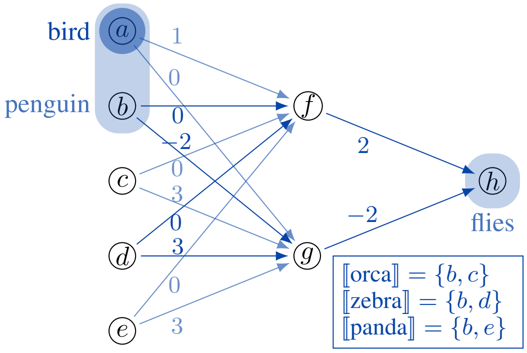
bird
a
1
0
penguin
b
0
-2
c
0
3
d
0
3
e
0
3
f
g
2
-2
h
flies
$[\![\text{orca}]\!] = \{b, c\}$
$[\![\text{zebra}]\!] = \{b, d\}$
$[\![\text{panda}]\!] = \{b, e\}$

Figure 3: 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. TODO

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 Hebb.

Theorem 3. The rules and axioms above are sound, i.e. hold for all $\mathcal{N} \in \mathsf{Net}$.

Proof. **FILL IN** □

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 3.

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	⟨11000⟩	orca	⟨01100⟩
zebra	⟨01010⟩	panda	⟨01001⟩

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 3. 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.

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. C. Kisby was supported in part by the US Department of Defense [Contract No. W52P1J2093009].

References

Martín Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

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

Christian Balkenius and Peter Gärdenfors. Nonmonotonic Inferences in Neural Networks. In *KR*, pages 32–39, 1991.

Alexandru Baltag, Nina Gierasimczuk, Aybüke Özgün, Ana Lucia Vargas Sandoval, and Sonja Smets. A dynamic logic for learning theory. *Journal of Logical and Algebraic Methods in Programming*, 109:100485, 2019a.

Alexandru Baltag, Dazhu Li, and Mina Young Pedersen. On the right path: a modal logic for supervised learning. In *International Workshop on Logic, Rationality and Interaction*, pages 1–14. Springer, 2019b.

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

Artur SD'Avila Garcez, Luis C Lamb, and Dov M Gabbay. *Neural-symbolic cognitive reasoning*. Springer Science & Business Media, 2008.

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

Laura Giordano, Valentina Gliozzi, and Daniele Theseider Dupré. From common sense reasoning to neural network models through multiple preferences: An overview. *CoRR*, abs/2107.04870, 2021. URL https://arxiv.org/abs/2107.04870.

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

Caleb Kisby, Saúl Blanco, and Lawrence Moss. The logic of hebbian learning. In *The International FLAIRS Conference Proceedings*, volume 35, 2022.

Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence*, 44(1-2):167–207, 1990.

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

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

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

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

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. Neuro-symbolic artificial intelligence: Current trends. *arXiv preprint arXiv:2105.05330*, 2021.

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

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

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