Neural Network Semantics

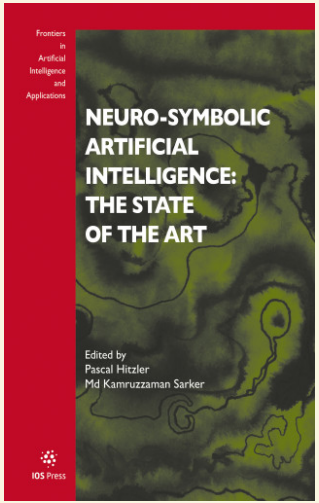
Thesis Proposal

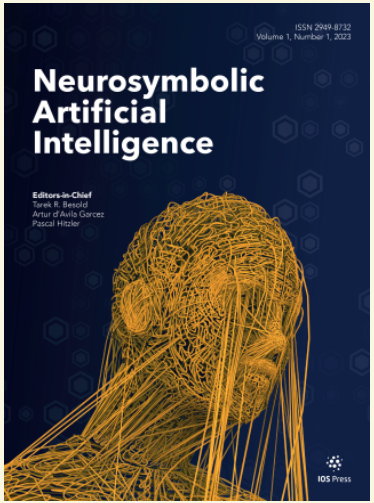
Caleb Schultz Kisby

December 17, 2024

The Neuro-Symbolic Question

How can we better understand and guide neural networks using logic?





Interfacing Logic + Neural Network Learning

- In the preface to the recent neuro-symbolic survey book, Frank van Harmelen writes:[1]

> What are the possible interactions between knowledge and learning? Can reasoning be used as a symbolic prior for learning ... Can symbolic constraints be enforced on data-driven systems to make them safer? Or less biased? Or can, vice versa, learning be used to yield symbolic knowledge? ... neuro-symbolic systems currently lack a theory that even begins to ask these questions, let alone answer them.

- My thesis work is on the interface between logic and neural networks, and *especially* logic and neural network learning!

1. van Harmelen, F. Preface: The 3rd AI Wave Is Coming, and It Needs a Theory. In In: Neuro-Symbolic Artificial Intelligence, IOS Press, 2022.

Thesis Statement

Neural networks can be treated as a class of models in formal logic (neural network semantics). In the course of developing these semantics, foundational questions about neural network inference and learning become natural and answerable questions in logic.

In particular:

Soundness	answers	"How can we formally verify that a class of neural networks and its learning policies obey certain properties?"
Completeness	answers	"How can we build a neural network that aligns with constraints, even as the net learns and changes over time?"
Expressivity	answers	"What kinds of functions and learning policies are neural networks capable of representing?"

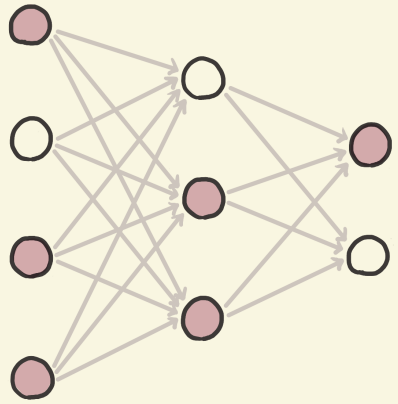
How do I defend this thesis?

Neural networks can be treated as a class of models in formal logic (neural network semantics). In the course of developing these semantics, foundational questions about neural network inference and learning become natural and answerable questions in logic.

- Of course, I need to explain *why* soundness, completeness, expressivity correspond to these questions.
- The word answerable here carries a lot of weight!
  - I need to actually *answer* these questions! i.e. prove soundness, completeness, expressivity results.
- Note the words inference *and* learning.
  - What makes my work novel is largely that I prove these results for a (simple) neural network *learning operator*.

Neural Network Semantics (Overview)

- Think of neural networks as *models* that interpret the logic
- Formulas φ will be mapped to *states* of the net, e.g.



- φ is *true at node* n iff φ *activates node* n
- I will use a dynamic modal logic, where
  - $\langle \boldsymbol{T} \rangle \varphi$ is interpreted as the *spread* of the signal/state φ
  - Dynamic operators $[\varphi]\psi$ model the *effects of learning/update*

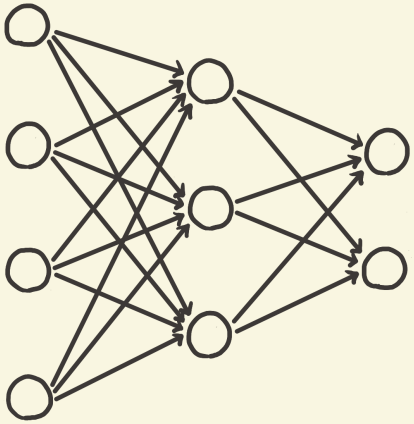
Why Modal Logic?

- I will use neural networks to interpret the dynamic modal language:

$$\varphi \in p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle \boldsymbol{K} \rangle \varphi \mid \langle \mathbf{K}^{\downarrow} \rangle \varphi \mid \langle \boldsymbol{T} \rangle \varphi \mid [P]\psi$$

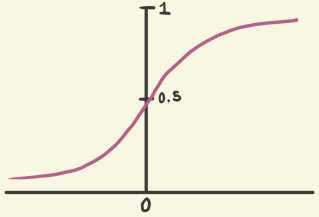
- But why interpret nets over **modal** logic? Three reasons:
  1. Modal logic is rich enough to express neural network *inference*.
     Intuition: I will interpret $\langle \boldsymbol{T} \rangle \varphi$ so that $\boldsymbol{T}\varphi \to \psi$ reads "ψ is activated by input φ", alternatively "the net classifies φ as ψ"
  2. It's easier to lift modal logic to a dynamic setting
     - My main goal is to study the effects of learning/update
  3. **An upsetting reason:** It's an open problem to find appropriate neural semantics for first-order logic!

Neural Network Semantics (The Details)

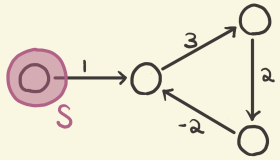


SAFE

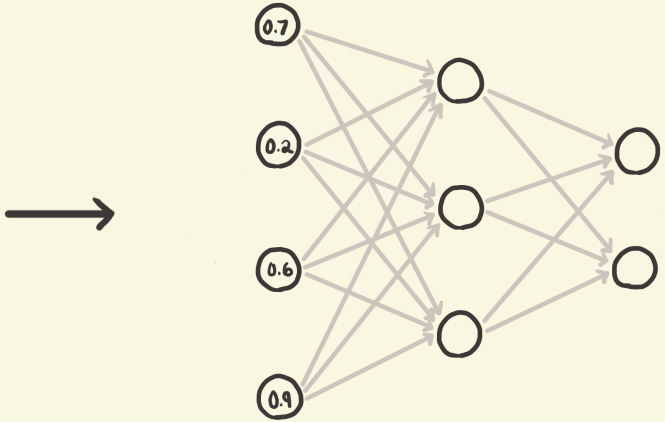
Binary (or “Saturated”)



Terminating

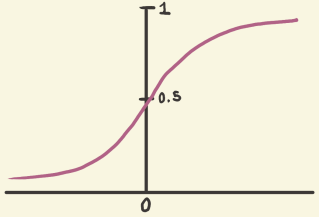


Neural Network Semantics (The Details)

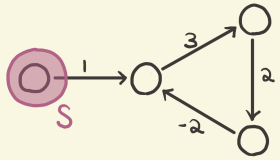


SAFE

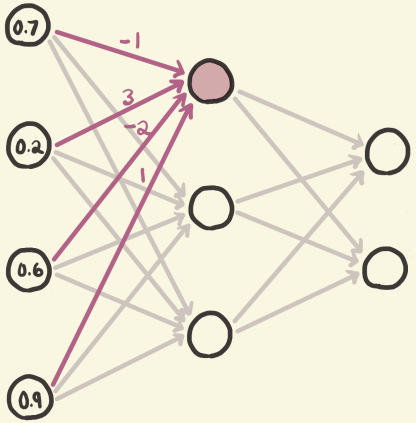
Binary (or “Saturated”)



Terminating

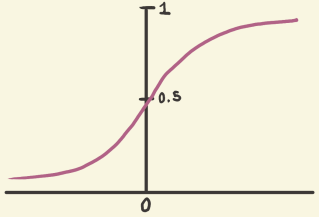


Neural Network Semantics (The Details)

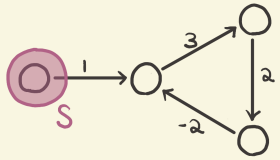


SAFE

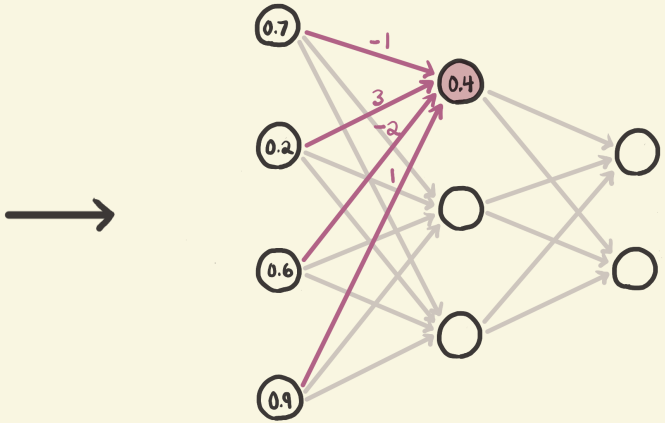
Binary (or “Saturated”)



Terminating

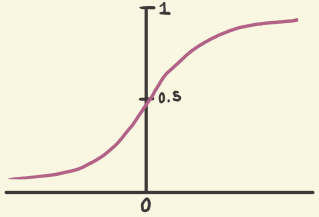


Neural Network Semantics (The Details)

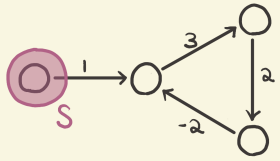


SAFE

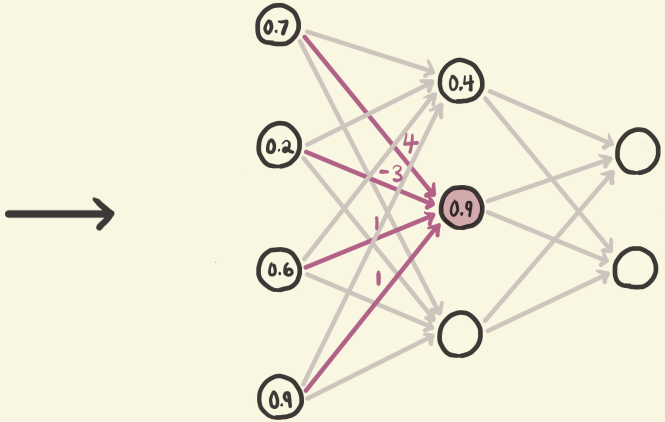
Binary (or “Saturated”)



Terminating

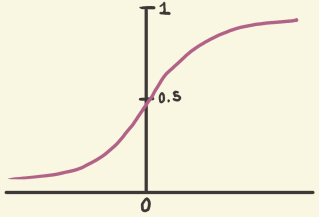


Neural Network Semantics (The Details)

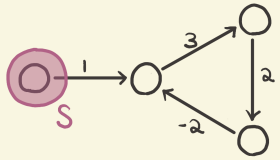


SAFE

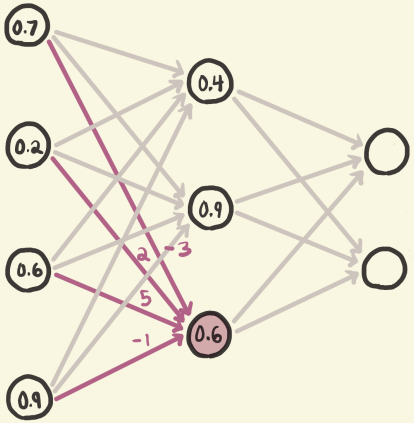
Binary (or “Saturated”)



Terminating

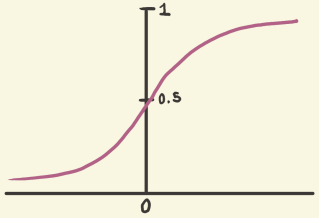


Neural Network Semantics (The Details)

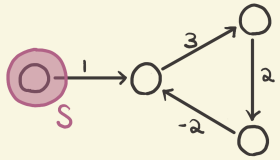


SAFE

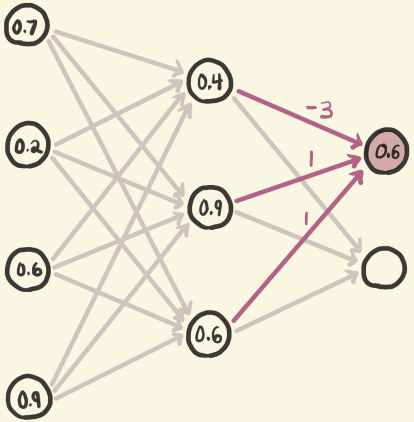
Binary (or "Saturated")



Terminating

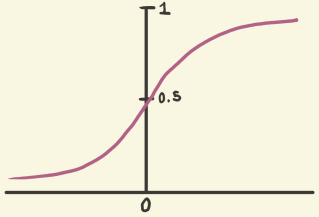


Neural Network Semantics (The Details)

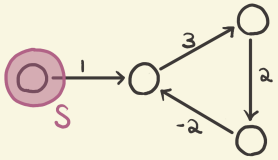


SAFE

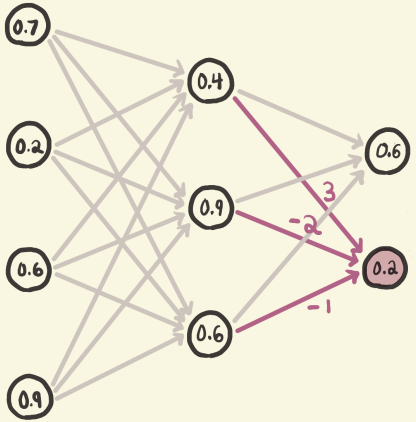
Binary (or “Saturated”)



Terminating

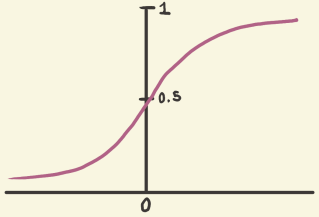


Neural Network Semantics (The Details)

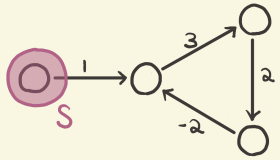


SAFE

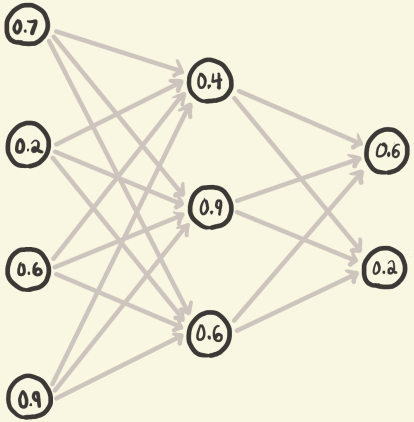
Binary (or “Saturated”)



Terminating

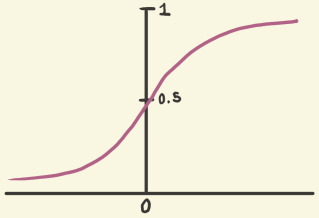


Neural Network Semantics (The Details)

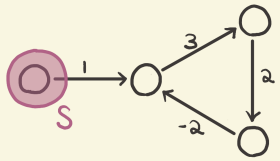


SAFE

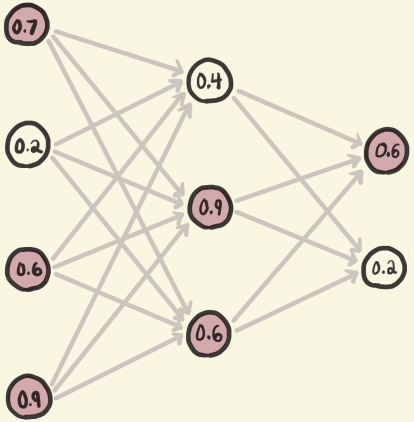
Binary (or “Saturated”)



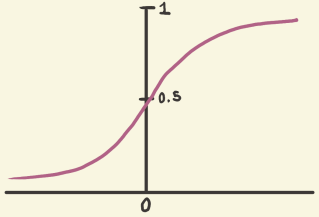
Terminating



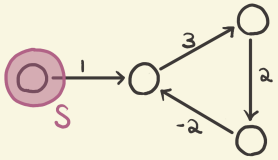
Neural Network Semantics (The Details)



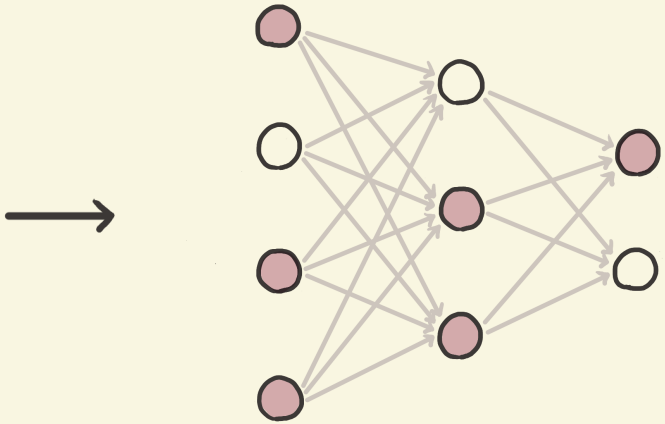
Binary (or “Saturated”)



Terminating

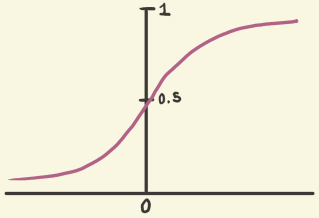


Neural Network Semantics (The Details)

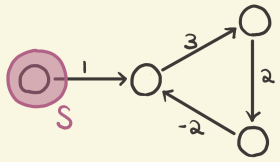


SAFE

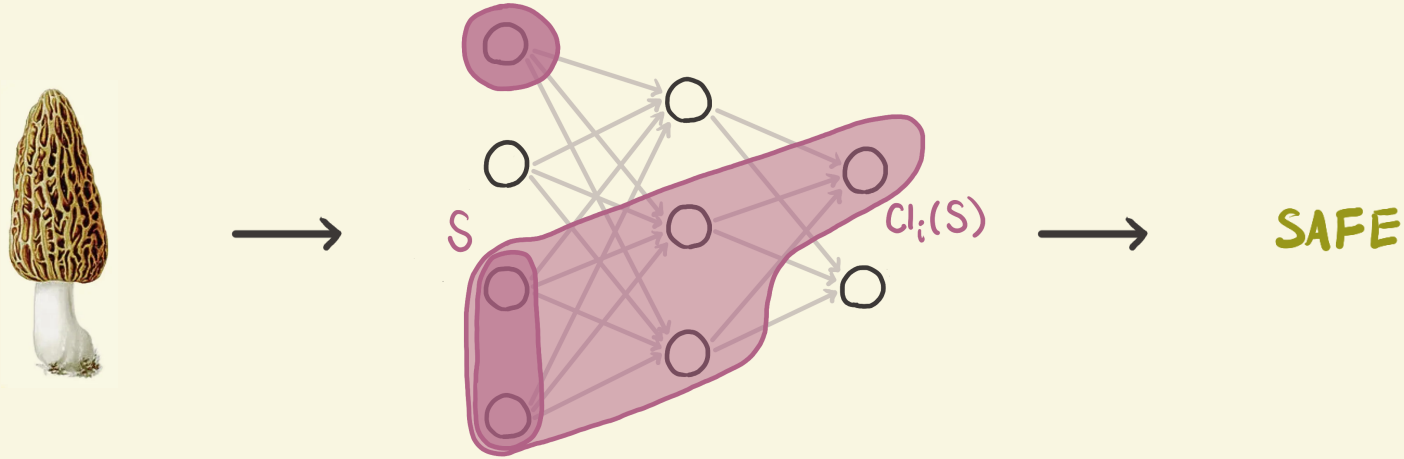
Binary (or “Saturated”)



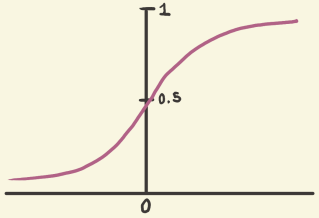
Terminating



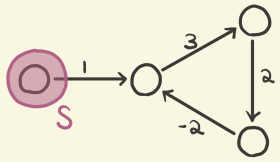
Neural Network Semantics (The Details)



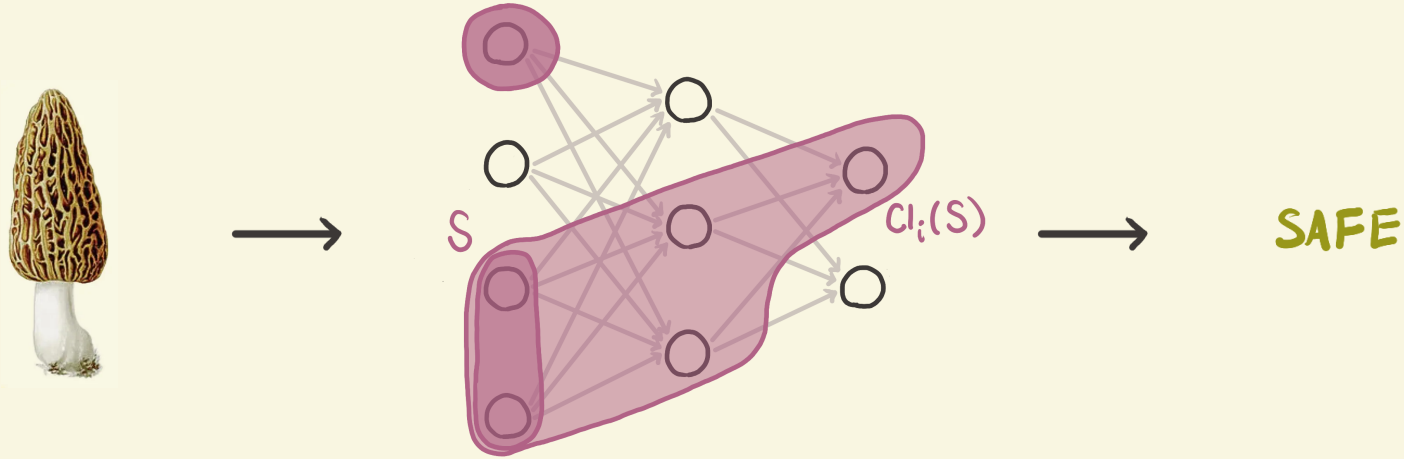
Binary (or "Saturated")



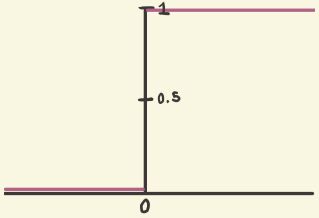
Terminating



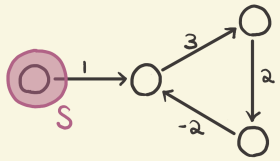
Neural Network Semantics (The Details)



Binary (or "Saturated")



Terminating



Neural Network Semantics (The Details)

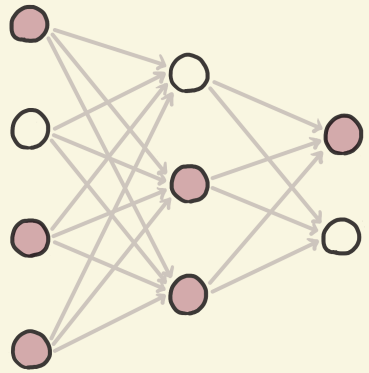
- A neural network model is $\mathcal{N} = \langle N, \text{bias}, E, W, A, \eta, V \rangle$
  - N is the set of nodes, bias a "bias" node, V : Prop $\rightarrow \mathcal{P}(N)$ a valuation,
  - E edges, $W(u, v)$ are weights, $A : \mathbb{Q} \rightarrow \{0, 1\}$ the activation function
  - η the learning rate (we will see later)

- A **state** is a possible activation pattern of the net: active (1) or not (0)
  The bias node is always active!

$$\text{State} = \{S \subseteq N \mid \text{bias} \in S\}$$

- Given an initial state S_0, each choice of E, W, A specifies a transition function F_{S_0} from State $\rightarrow$ State:

$$F_{S_0}(S) = S_0 \cup \left\{ n \mid A\left(\sum_{m \in \text{preds}(n)} W(m,n) \cdot \chi_S(m) \right) = 1 \right\}$$



Neural Network Semantics (More Details!)

- Moreover, I assume the net is **terminating**! i.e. for all $S_0 \in \text{State}$, F_{S_0} applied repeatedly to S_0, i.e.

  $$S_0, F_{S_0}(S_0), F_{S_0}(F_{S_0}(S_0)), \ldots, F_{S_0}^k(S_0), \ldots$$

  eventually terminates at some least fixed point. Let $\mathbf{Net}$ be the class of all such neural network models.

- Let the **closure** $\text{Cl}(S) : \text{State} \to \text{State}$ be this fixed point.
  - $\text{Cl}(S)$ is what I mean by the *spread* (or diffusion, or forward propagation) of the signal S through the net.

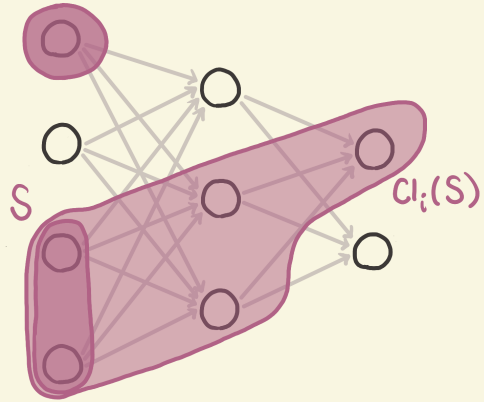
- I will also need "helper" functions for graph-reachability:
  - $\text{Reach}(S)$ and $\text{Reach}^{\downarrow}(S)$ for expressing graph reachability.

Neural Network Semantics (Even More Details!!)

Finally, the semantics:

For all $\mathcal{N} \in \mathbf{Net}$, $n \in N$:

$$
\begin{array}{lll}
\mathcal{N}, n \Vdash p & \text{iff} & n \in V(p) \\
\mathcal{N}, n \Vdash \neg\varphi & \text{iff} & \mathcal{N}, n \nVdash \varphi \\
\mathcal{N}, n \Vdash \varphi \wedge \psi & \text{iff} & \mathcal{N}, n \Vdash \varphi \text{ and } \mathcal{N}, n \Vdash \psi \\
\mathcal{N}, n \Vdash \langle \boldsymbol{K} \rangle \varphi & \text{iff} & n \in \mathsf{Reach}^{\downarrow}([\![\varphi]\!]) \\
\mathcal{N}, n \Vdash \langle \mathbf{K}^{\downarrow} \rangle \varphi & \text{iff} & n \in \mathsf{Reach}([\![\varphi]\!]) \\
\mathcal{N}, n \Vdash \langle \boldsymbol{T} \rangle \varphi & \text{iff} & n \in \mathsf{Cl}([\![\varphi]\!])
\end{array}
$$



where $[\![\varphi]\!] = \{n \mid \mathcal{N}, n \Vdash \varphi\}$

- the nodes “activated by” φ

Dynamic Update in Neural Networks

- Inspired by Dynamic Epistemic Logic (DEL)[2]
- We interpret $[P]\varphi$ as "after updating our model by P, φ holds"

$$\mathcal{N}, n \Vdash [P]\varphi \quad \text{iff} \quad \text{Update}(\mathcal{N}, [\![P]\!]), n \Vdash \varphi$$

- **Single-Step Hebbian Update:**
  - "Neurons that fire together wire together"

$$\text{Hebb}(\langle N, \text{bias}, E, W, A, \eta, V\rangle,\ S) = \langle N, \text{bias}, E, W', A, \eta, V\rangle$$

  - where we (1) take the nodes $\text{Cl}([\![\varphi]\!])$ activated by $[\![\varphi]\!]$, and (2) increase the involved weights by the learning rate $\eta \geq 0$:

$$W'(m,n) = W(m,n) + \eta \cdot \chi_{\text{Cl}(S)}(m) \cdot \chi_{\text{Cl}(S)}(n)$$

2. van Ditmarsch, H., van Der Hoek, W., and Kooi, B. *Dynamic epistemic logic*. Springer, 2007.

Dynamic Update in Neural Networks (Contd)

- **Iterated Hebbian Update:**
  - What happens when we repeatedly update $\mathcal{N}$ by S, until the weights within S become saturated?

$$\text{Hebb}^{\star}(\langle N, \text{bias}, E, W, A, \eta, V\rangle,\ S) = \langle N, \text{bias}, E, W', A, \eta, V\rangle$$

  - where this time we repeatedly increase the involved weights an iter number of times:

$$W'(m,n) = W(m,n) + \text{iter} \cdot \eta \cdot \chi_{\text{Cl}(S)}(m) \cdot \chi_{\text{Cl}(S)}(n)$$

  - This is a very simple update, but I have the complete story for it!

Back to the 3 Questions

- **Soundness**
  - **Inference:** What axioms are sound for $\langle \boldsymbol{K} \rangle \varphi, \langle \mathbf{K}^{\downarrow} \rangle \varphi, \langle \boldsymbol{T} \rangle \varphi$?
  - **Learning:** What axioms are sound for $[P]_{\text{Hebb}} \varphi, [P]_{\text{Hebb}^\star} \varphi$?
  - i.e. so that if $\vdash \varphi$ then $\mathcal{N}, n \Vdash \varphi$ for all $\mathcal{N}$ and all n

- **Completeness**
  - **Inference:** Can we do model building for $\langle \boldsymbol{K} \rangle \varphi, \langle \mathbf{K}^{\downarrow} \rangle \varphi, \langle \boldsymbol{T} \rangle \varphi$?
  - **Learning:** Can we do model building for $[P]_{\text{Hebb}^\star} \varphi$?
  - i.e. for all consistent sets Γ can we build a net $\mathcal{N}, n \Vdash \Gamma$ for all n?

- **Expressivity**
  - **Inference:** What operators can $\mathbf{Net}$ simulate, and vice-versa?
  - **Learning:** What *updates* can $\mathbf{Net} + [P]_{\text{Hebb}^\star} \varphi$ simulate, and vice-versa?

Progress and Goals

	Static Inference – Reach and CI	Dynamic Update – Hebb*
Soundness	✓	✓
Completeness	in progress	✓
Expressivity	in progress	in progress

Progress and Goals: Soundness

- **Soundness**
  - **Inference:** What axioms are sound for $\langle \boldsymbol{K} \rangle \varphi, \langle \mathbf{K}^{\downarrow} \rangle \varphi, \langle \boldsymbol{T} \rangle \varphi$?
  - **Learning:** What axioms are sound for $[P]_{\text{Hebb}} \varphi, [P]_{\text{Hebb}^{\star}} \varphi$?

- ✅ It's easy to list sound axioms for $\langle \boldsymbol{K} \rangle \varphi, \langle \mathbf{K}^{\downarrow} \rangle \varphi, \langle \boldsymbol{T} \rangle \varphi$
  - $\langle \boldsymbol{T} \rangle \varphi$ follows axioms drawn from Leitgeb's work[3]
- ✅ In my work, I have given sound axioms for $[P]_{\text{Hebb}} \varphi$[4] (and later $[P]_{\text{Hebb}^{\star}} \varphi$)
- ✅ I also wrote a Python program which does neural net model checking based on these sound semantics:

https://github.com/ais-climber/a-la-mode

3. Leitgeb, H. Nonmonotonic reasoning by inhibition nets. In Artificial Intelligence, 2001.

4. Schultz Kisby, C., Blanco, S., and Moss, L. The Logic of Hebbian Learning. The International FLAIRS Conference Proceedings, 2022.

Progress and Goals: Completeness

- **Completeness**
  - **Inference:** Can we do model building for $\langle K \rangle \varphi$, $\langle \mathbf{K}^{\downarrow} \rangle \varphi$, $\langle T \rangle \varphi$?
  - **Learning:** Can we do model building for $[P]_{\text{Hebb}^\star} \varphi$?

- ✓ In a recent paper[5], I showed that we can actually translate $[P]_{\text{Hebb}^\star} \varphi$ into the base language via reduction axioms.
- ✓ As a bonus, I checked the major results in this paper using Lean: https://github.com/ais-climber/argyle
- I have been working on modifying Leitgeb's construction for $\langle K \rangle \varphi$, $\langle \mathbf{K}^{\downarrow} \rangle \varphi$, $\langle T \rangle \varphi$, and I'm pretty confident I can do it.
- I would also like to write this model building into the Python program

5. Schultz Kisby, C., Blanco, S., and Moss, L. What Do Hebbian Learners Learn? Reduction Axioms for Iterated Hebbian Learning. Proceedings of AAAI, 2024.

Progress and Goals: Expressivity

- **Expressivity**
  - **Inference:** What operators can $\mathbf{Net}$ simulate, and vice-versa?
  - **Learning:** What *updates* can $\mathbf{Net}+[P]_{\text{Hebb}^\star}\varphi$ simulate, and vice-versa?

- Many open questions, I've been wrapping my head around this for ~1 year
- ✓ $\mathbf{Net}$ can simulate graph models $\mathbf{Rel}$, plausibility models $\mathbf{Plaus}$, and social majority models. It can be simulated by neighborhood models.
- Can $\mathbf{Plaus}$ simulate $\mathbf{Net}$? I have a non-constructive proof, but I would like a model building construction!
- What $\mathbf{Net}$ updates are equivalent to $\mathbf{Plaus}$ updates, and vice-versa? Can Hebb* simulate plausibility re-orderings such as Lex, Consr?[6]

6. van Benthem, J. Dynamic logic for belief revision. Journal of Applied Non-Classical Logics, 2007.

Closing Thoughts

- I am not the originator of neural network semantics; my work is novel because I prove the first ever soundness, completeness, and expressivity results for a neural network update (in this case, Hebb*).
  - See my proposal write-up for a short history of this approach, and related work that I neglected to mention here.

- I have made many simplifying assumptions about neural networks (binary, must have bias node, non-oscillating), about the learning (Hebbian, unstable), and about the constraint language (modal)
  A long-term goal of mine is to do this kind of analysis for neural networks used in practice (fuzzy, with backpropagation, possibly for transformer models), in a richer constraint language (FOL).
  - I recently submitted a DFF proposal with Prof. Nina Gierasimczuk to take this work in these directions.

Thank you for your attention!

	Static Inference – Reach and CI	Dynamic Update – Hebb*
Soundness	✓	✓
Completeness	in progress	✓
Expressivity	in progress	in progress

Contact:
Caleb Schultz Kisby
cckisby@gmail.com
https://ais-climber.github.io/