Neural Network Semantics: Project Summary

Caleb Schultz Kisby, Saúl Blanco, and Lawrence Moss
Luddy School of Informatics, Computing, and Engineering

SCALE

Neural Network Semantics

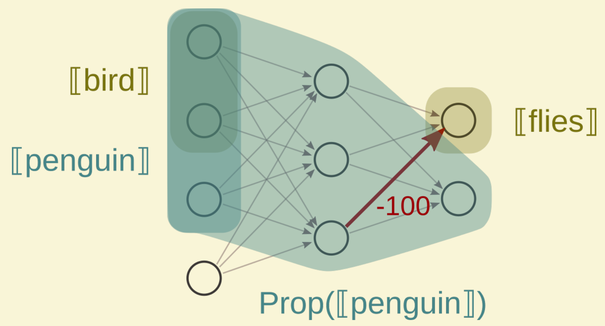
Definition. The neural networks N we consider are weighted, fully-connected, terminating nets with binary activation functions. The net's states (activation patterns) are just given by sets of nodes.

Definition. The forward-propagation $\mathrm{Prop}(S)$ gives the set of nodes that are eventually activated by S. Think of this as the "spread" of S throughout the net.

Key Idea: Neural networks are not merely black boxes! $\mathrm{Prop}(S)$ contains information about conditional beliefs:

Let's say $A \Rightarrow B$ holds iff $\mathrm{Prop}([\![A]\!]) \supseteq [\![B]\!]$; in other words, the net *classifies* A as B. (Leitgeb 2018) shows that we can build a neural network (with states) satisfying a set of conditional constraints Γ.

Example. Let $\Gamma = \{\text{penguins} \to \text{bird}, \text{bird} \Rightarrow \text{flies}, \neg(\text{penguins} \Rightarrow \text{flies})\}$. Here's how we might build N:



Syntax. $A, B \in p \mid \neg A \mid A \wedge B \mid \mathbf{K}A \mid \mathbf{T}A$

We define the duals $\langle \mathbf{K} \rangle, \langle \mathbf{T} \rangle$ as usual. We can express $A \Rightarrow B$ as $\mathbf{T}A \to B$ ("the typical A is B").

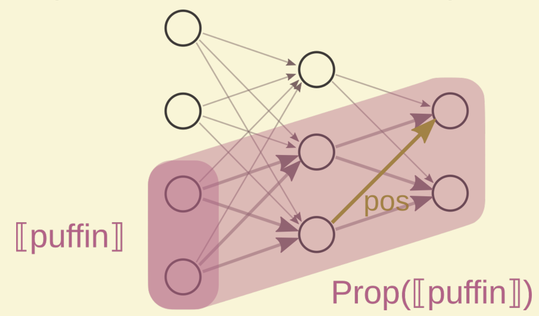
Semantics. First, each atomic concept p is mapped to a state $V(p)$. We then interpret the rest of the language directly on the neural network itself:

$N, n \Vdash p$	iff	$n \in V(p)$
$N, n \Vdash \langle \mathbf{K} \rangle A$	iff	n is graph-reachable from A
$N, n \Vdash \langle \mathbf{T} \rangle A$	iff	$n \in \mathrm{Prop}(\{N, u \Vdash A\})$

Modeling Neural Network Learning

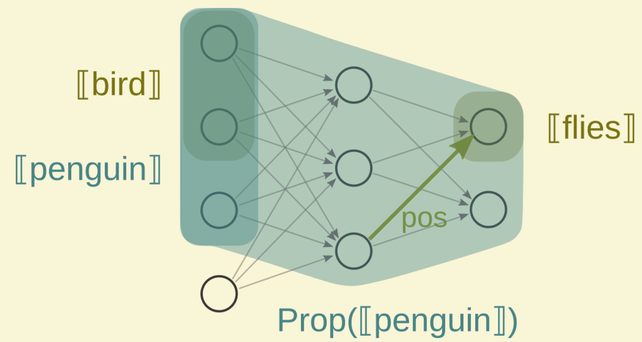
These semantics don't account for learning! e.g., Consider iterated Hebbian learning, which says

Neurons that fire together wire together;
Repeat until we reach a fixed point.



Definition. $\mathrm{Hebb}^*(N, [\![S]\!])$ gives the resulting net obtained by increasing the weights of N within $\mathrm{Prop}([\![S]\!])$ until they are "maximally high."

Example. Say the neural network we built before repeatedly observes puffins (shown in the above picture). Puffins share enough features with penguins that the net eventually believes that penguins fly.



Learning wrecks the model! How can we track the precise way in which the network model changes?

We can model this logically via dynamic formulas $[A]B$ (read "after learning A, B holds"). Formally,

$$[\![[A]B]\!]_N \;=\; [\![B]\!]_{\mathrm{Hebb}^*(N, [\![A]\!])}$$

Can we completely characterize $[A]$'s effect on the net?

Key Results

Soundness for Learning. We proved sound laws that hold for all nets using iterated Hebbian learning.

Completeness for Learning. We found a complete characterization of iterated Hebbian learning. The central axiom states:

$$[A]\mathbf{T}B \leftrightarrow \mathbf{T}([A]B \wedge (\mathbf{T}A \vee \mathbf{K}(\mathbf{T}A \vee \mathbf{T}[A]B)$$

Model Building. We demonstrated a technique for building a neural network that satisfies constraints on the net's behavior *after learning*!

Formal Verification. Our key results were formally verified using the Lean proof assistant.

Publications:

- Caleb Schultz Kisby, S. Blanco, and L. Moss. What Do Hebbian Learners Learn? Reduction Axioms for Iterated Hebbian Learning. AAAI 2024.
- Caleb Kisby, S. Blanco, and L. Moss. The Logic of Hebbian Learning. FLAIRS 2022. *Nominated for Best Student Paper.*

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

Future Work

- Can we extend this to more sophisticated learning policies? Consider: convergence, supervised learning, backpropagation, single-step update ...
- How can we use this in practice to constrain nets throughout their training? (AI Alignment)

Thanks and Contact

This work was funded in part by the US Department of Defense [Contract No. W52P1J2093009], through the NSWC Crane SCALE program. **Thank you for your support!**

Contact: Caleb Schultz Kisby – `cckisby@iu.edu`