Reduction Axioms for Iterated Hebbian Learning

Caleb Schultz Kisby, Saúl Blanco, and Lawrence Moss
Indiana University

Neural Network Semantics

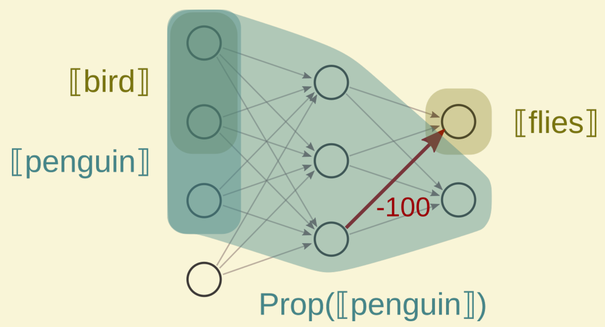
Definition. The neural networks N we consider are weighted, fully-connected, feed-forward nets with binary activation functions. The net's states (activation patterns) are just given by sets of nodes.

Definition. The forward-propagation $\mathrm{Prop}(S)$ gives the set of nodes that are eventually activated by S.

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. We consider the language:

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

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. We map each formula to a state:

$$[\![p]\!] = V(p) \quad [\![\neg A]\!] = [\![A]\!]^{\complement} \quad [\![A \wedge B]\!] = [\![A]\!] \cap [\![B]\!]$$

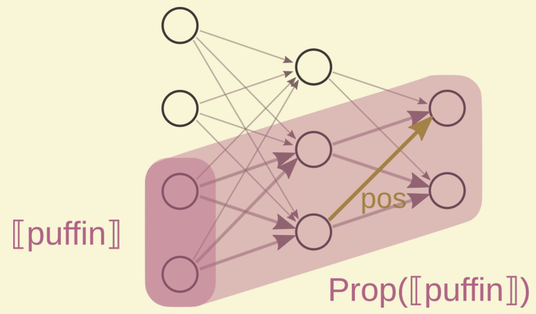
$$\begin{aligned} [\![\langle \mathbf{K} \rangle A]\!] &= \{n \mid n \text{ is graph-reachable from } A\} \\ [\![\langle \mathbf{T} \rangle A]\!] &= \mathrm{Prop}([\![A]\!]) \end{aligned}$$

Definition. $N, w \vDash A$ iff $w \in [\![A]\!]$

Iterated Hebbian Learning

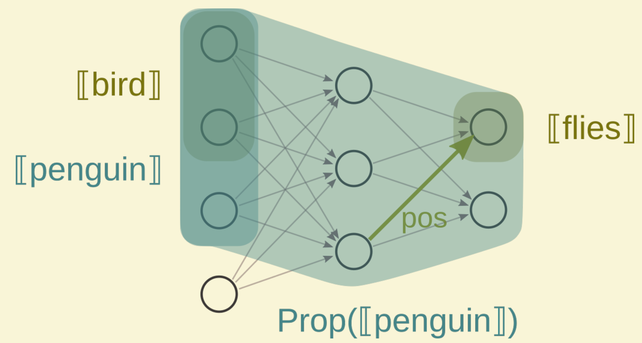
These semantics don't account for learning! e.g., Consider iterated Hebbian learning, which says

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?

Main Results

Theorem. The following axioms are sound:

$$\begin{aligned} [A]p &\leftrightarrow p \\ [A]\neg B &\leftrightarrow \neg [A]B \\ [A](B \wedge C) &\leftrightarrow [A]B \wedge [A]C \\ [A]\mathbf{K}B &\leftrightarrow \mathbf{K}[A]B \\ [A]\mathbf{T}B &\leftrightarrow \mathbf{T}([A]B \wedge (\mathbf{T}A \vee \mathbf{K}(\mathbf{T}A \vee \mathbf{T}[A]B)) \end{aligned}$$

Theorem. *Assuming* model building for the base language, for all consistent $\Gamma \subseteq L$ there is a net N such that $N \vDash \Gamma$

Theorem. *Assuming* completeness for the base language, $[A]$ is completely axiomatized by the reduction axioms above.

Future Work

- Can we extend this to more sophisticated learning policies? Consider: convergence, supervised learning, single-step update ...
- Could we do this analysis for backpropagation?
- How can we use this in practice to constrain nets throughout their training? (AI Alignment)
- What is the relationship between neural network learning and plausibility upgrade?

Thanks and Contact

This work was funded in part by the US Department of Defense [Contract No. W52P1J2093009]. Thanks as well to the anonymous reviewers for their helpful feedback and suggestions!

Contact: Caleb Schultz Kisby
cckisby@iu.edu
ais-climber.github.io