Logics with Neural Network Semantics

based on my dissertation (coming soon!)

Caleb Schultz Kisby @ Indiana University

Introduction

Assume: All neural networks N have binary activations. Nets may be recurrent, but every activated state S must eventually stabilize to a unique fixed point.

Definition. The closure $\mathrm{Clos}(S)$ over N gives this unique fixed point—the set of nodes that are activated by S.

Think of this as the "spread" of S throughout the net.

Key Idea: We can interpret logics directly over neural networks using $\mathrm{Clos}(S)$. Let's first look at two logics:

- A conditional logic $L_{\Rightarrow}$ whose conditional $\varphi \Rightarrow \psi$ is interpreted as saying the net *classifies* φ as ψ:

$$N \vDash \varphi \Rightarrow \psi \quad \text{iff} \quad \mathrm{Clos}(\llbracket\varphi\rrbracket) \supseteq \llbracket\psi\rrbracket$$

- A non-normal modal logic $L_{\mathbf{C}}$ whose modality $\langle\mathbf{C}\rangle\varphi$ is interpreted as the closure $\mathrm{Clos}(\llbracket\varphi\rrbracket)$ in the neural net:

$$N, n \Vdash \langle\mathbf{C}\rangle\varphi \quad \text{iff} \quad n \in \mathrm{Clos}(\llbracket\varphi\rrbracket)$$

I also include two helper operators, $\langle\mathbf{K}\rangle\varphi$ and $\mathbf{A}\varphi$.

Note 1. $\llbracket\varphi\rrbracket$ is shorthand for $\{m \mid N, m \Vdash \varphi\}$

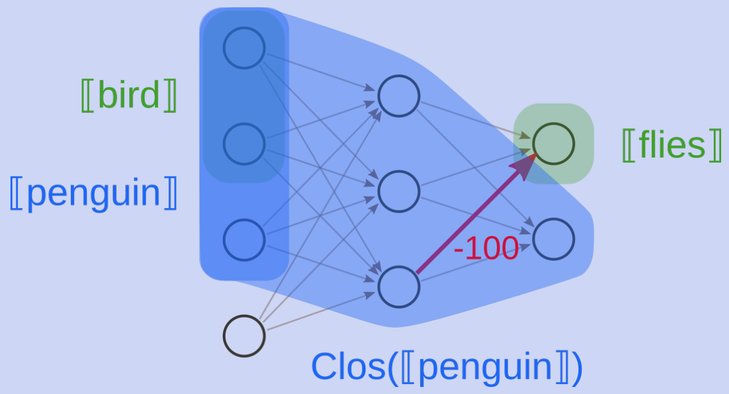
Note 2. $\mathbf{C}, \mathbf{K}, \mathbf{E}$ are the duals of $\langle\mathbf{C}\rangle, \langle\mathbf{K}\rangle, \mathbf{A}$

Note 3. $\mathbf{A}(\psi \to \langle\mathbf{C}\rangle\varphi)$ is the same as $\varphi \Rightarrow \psi$

An Example

Here is an example net that satisfies some logic constraints:

- penguins $\to$ bird (*all penguins are birds*)
- $\mathbf{A}(\text{flies} \to \langle\mathbf{C}\rangle\text{bird})$ (*classify birds as flying*)
- $\neg\mathbf{A}(\text{flies} \to \langle\mathbf{C}\rangle\text{penguin})$ (*don't classify peng. as flying*)



Soundness & Completeness

When a logic with neural network semantics is

- Sound: then we can formally verify that certain algebraic properties hold over this class of neural nets.
- Complete: then we can do model-building; for *any* finite set of constraints Γ over the logic, we can construct a neural network satisfying Γ.

Theorem. (Leitgeb 2003) $L_{\Rightarrow}$ has a sound and complete proof system $\vdash$ (the rules of cumulative conditional logic).

Theorem. $L_{\mathbf{C}}$ has a sound and complete proof system $\vdash$.

How do we do model-building? Design a canonical neural net, whose nodes are formulas α in "normal form."

- Every finite Γ is equivalent to some normal form α
- Pick edges, weights, and activation function so that

$$\alpha \in \mathrm{Clos}(\llbracket\varphi\rrbracket) \quad \text{iff} \quad \vdash \alpha \to \langle\mathbf{C}\rangle\varphi$$

Expressivity

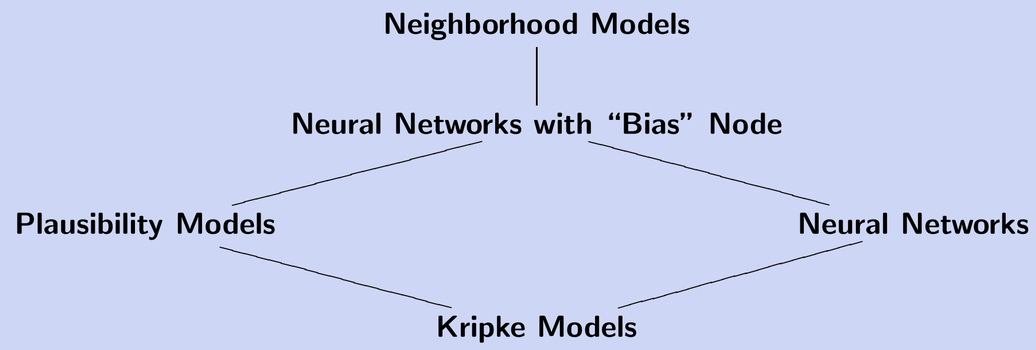
What classes of models can neural networks simulate?

Note: This is not quite the same as descriptive complexity ("What properties/functns. are definable over neural nets?")

Definition. A class of models C_2 can simulate C_1 if there is injective $f : C_1 \to C_2$ such that $\forall M \in C_1$ and $\forall \varphi \in L$,

$$M \vDash \varphi \quad \text{iff} \quad f(M) \vDash \varphi$$

Theorem. Over $L_{\mathbf{C}}$, we have the following lattice, where each class can simulate the models in the classes below it.

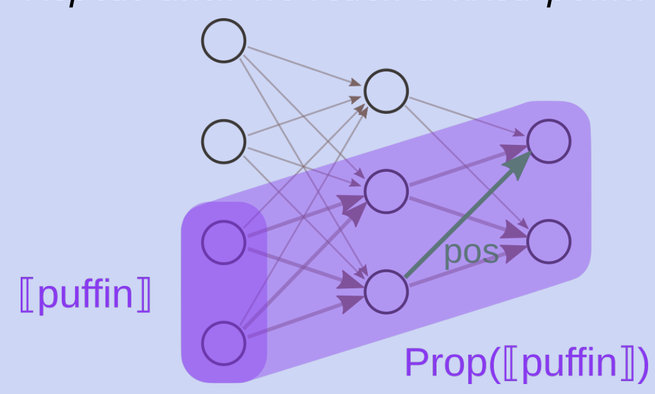


Neural Network Update

We can model neural network learning using dynamic logic! Add to $L_{\mathbf{C}}$ dynamic formulas $[\varphi]\psi$, which reads "ψ holds after a simple neural network update on input φ."

As a proof-of-concept, I've worked this out for a simple update. Consider strong Hebbian learning, which says

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



Let $\mathrm{Hebb}(N, S)$ be the net obtained by increasing the weights of N within $\mathrm{Clos}(S)$ until they are "maximally high."

We interpret $[\varphi]\psi$ in the semantics by:

$$N, n \Vdash [\varphi]\psi \quad \text{iff} \quad \mathrm{Hebb}(N, \llbracket\varphi\rrbracket), n \Vdash \psi$$

Assume: N is fully connected, activation is binary step

Theorem. $\mathrm{L}_{\mathrm{Hebb}}$ can be completely axiomatized by "translating away" $[\varphi]\psi$ over $L_{\mathbf{C}}$. The key reduction axiom is:

$$[\varphi]\mathbf{C}\psi \leftrightarrow \mathbf{C}([\varphi]\psi \wedge (\mathbf{C}\varphi \vee \mathbf{K}(\mathbf{C}\varphi \vee \mathbf{C}[A]\psi)$$

Future Work

- What is the relationship between these expressivity results and the FLaNN community's work on NN expressivity?
- Can we axiomatize gradient descent / backpropagation?
- Can we use neural network model building in practice to constrain nets throughout their training? (AI Alignment)

Dissertation Draft (in progress)**:**

caleb.schultzkisby.me/Research_Papers/Dissertation/dissertation.pdf

Contact: cckisby@gmail.com caleb.schultzkisby.me