Reasoning about Neural Network Learning

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

Reasoning about Static Nets

Monotonicity Axioms

know (A → B) → (know A → know B)
(typ A1 → A2) … (typ An → A1) →
  (typ Ai ↔ Aj)

Basic Modal Axioms

know A → A
know A → know know A
typ A → A
typ A → typ typ A
know A → typ A

Syntax	Classical Meaning	Neural Network
p	proposition	a (fuzzy) set of neurons
A **and** B	A and B	$A \cup B$
A → B	A implies B	$A \supseteq B$
know A	the agent knows A	the set of neurons reachable from A
typ A	typically A	the set of neurons activated by A
A ⇒ B	typ A → B	on input A the net predicts B
[**hebb** A] B	incremental pref upgrade on A	learn A (Hebbian)
[**hebb*** A] B	preference upgrade on A	repeatedly learn A (Hebbian)

Reasoning about Learning

Induction Axioms

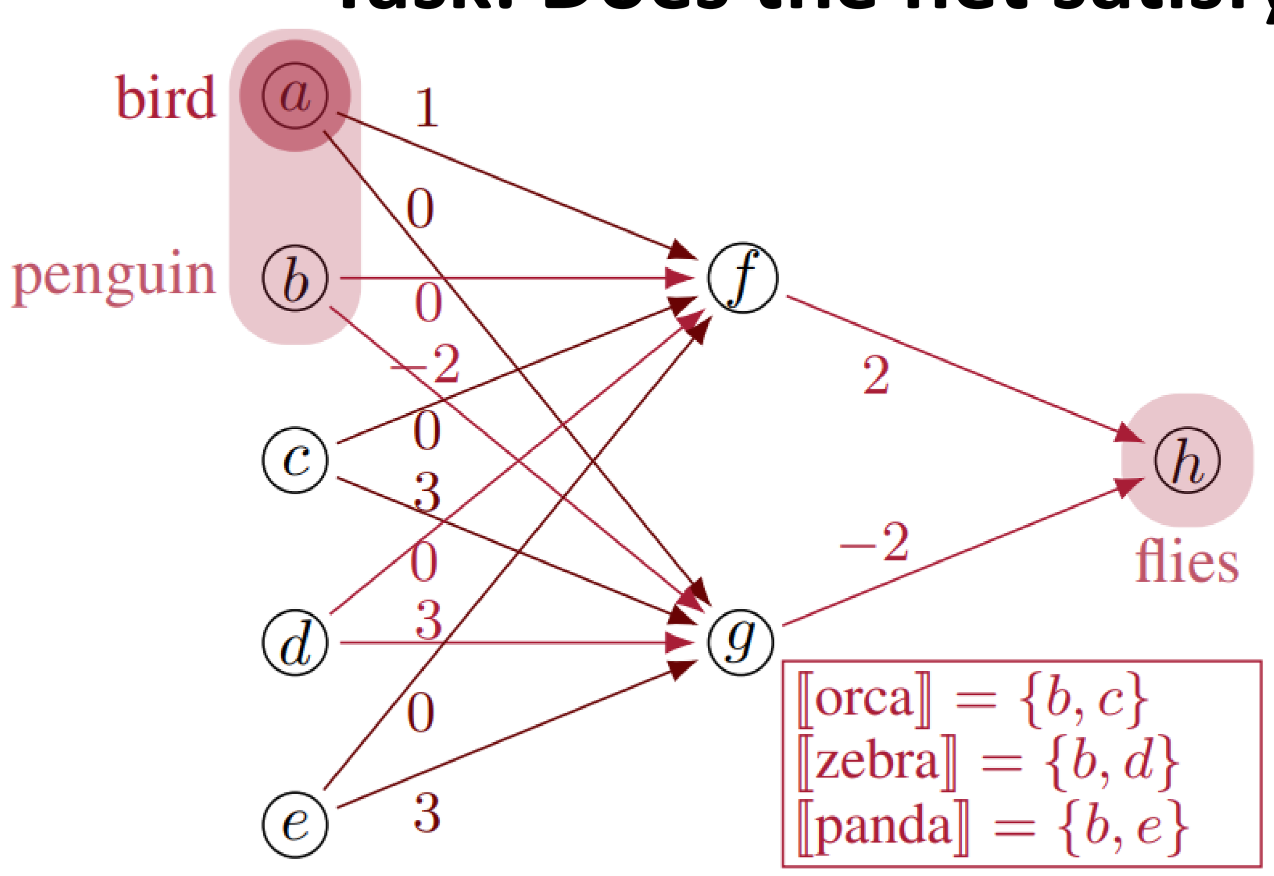
[hebb* A] B → B and [hebb A][hebb* A] B
[hebb* A](B → [hebb A] B) → [hebb* A] B

What The Net Learns

[hebb* A] typ B ↔
$\begin{cases} \textbf{typ [hebb* A] B} & \text{if typ A or typ B is } \emptyset \\ \textbf{typ [hebb* A] B and (typ A or know B)} & \text{otherwise} \end{cases}$

Model Checking

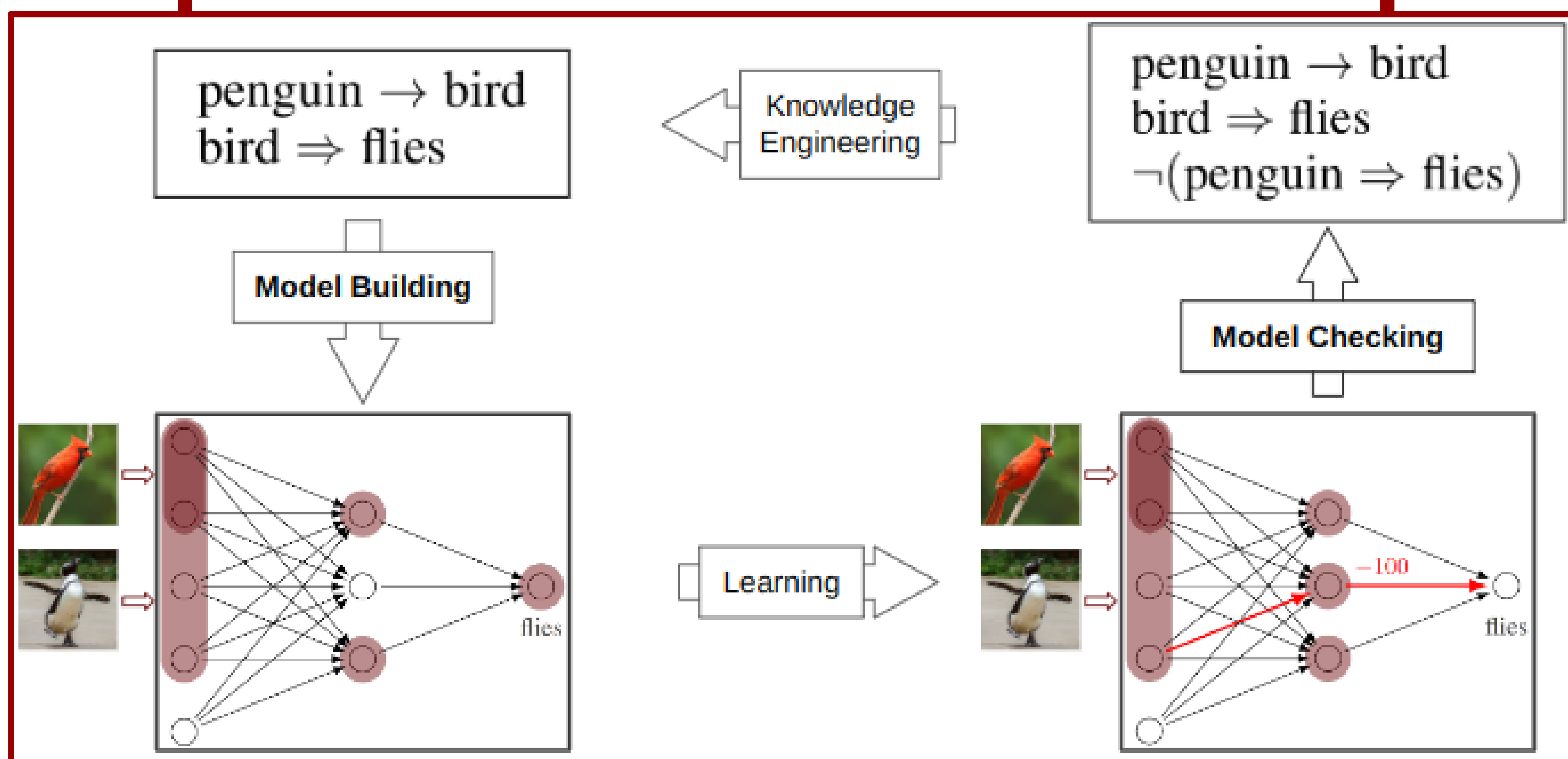
Task: Does the net satisfy P?



$\mathcal{N} \models$ typ penguin → flies, but
$\mathcal{N} \not\models$ [hebb orca] [hebb zebra] [hebb panda]
  typ penguin → flies

```
>>> print(model.is_model("typ penguin → flies"))
True

>>> print(model.is_model("[hebb orca] [hebb zebra] [hebb panda] \
                          typ penguin → flies))
False
```



github.com/ais-climber/neural-semantics

Model Building

Task: Build a net that satisfies P.

GOAL. (Binary, feedforward) nets are equivalent to a certain class of classical modal frames.

COROLLARY. Given a knowledge base Γ, we can construct a net $\mathcal{N}$ such that $\mathcal{N} \models \Gamma$

COROLLARY. The axioms for reasoning about know, typ, and [hebb* A] are complete.

Work in Progress

- Use Lean to verify model checking code
- Finish proof for model building
- Extend system to reason about fuzzy sets
- Extend with [**backprop** A] (backpropagation)

Acknowledgments: This work was funded by the US Department of Defense (Contract W52P1J2093009). Thank you for your support!