A Semantic Theory for Neuro-Symbolic AI

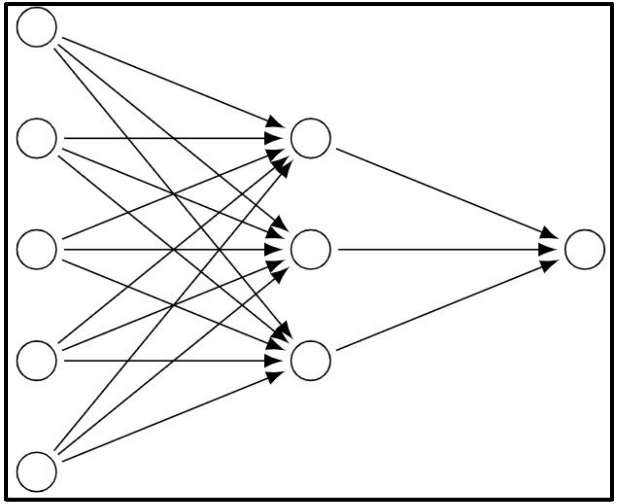
Caleb Schultz Kisby
w. Saúl Blanco & Larry Moss

INDIANA UNIVERSITY BLOOMINGTON

The Crisis in AI

$\neg(\text{airplane} \rightarrow \text{bird})$
$\mathbf{T}(\text{airplane} \rightarrow \text{flies})$
$\text{penguin} \rightarrow \text{bird}$
$\mathbf{T}(\text{bird}) \rightarrow \text{flies}$
$\neg(\text{penguin} \rightarrow \text{flies})$

Problem:
How can we reconcile the two?



(Modal) Logic

$$p \quad \bot \quad \neg A \quad A \to B \quad \mathbf{K}A \quad \mathbf{T}A$$

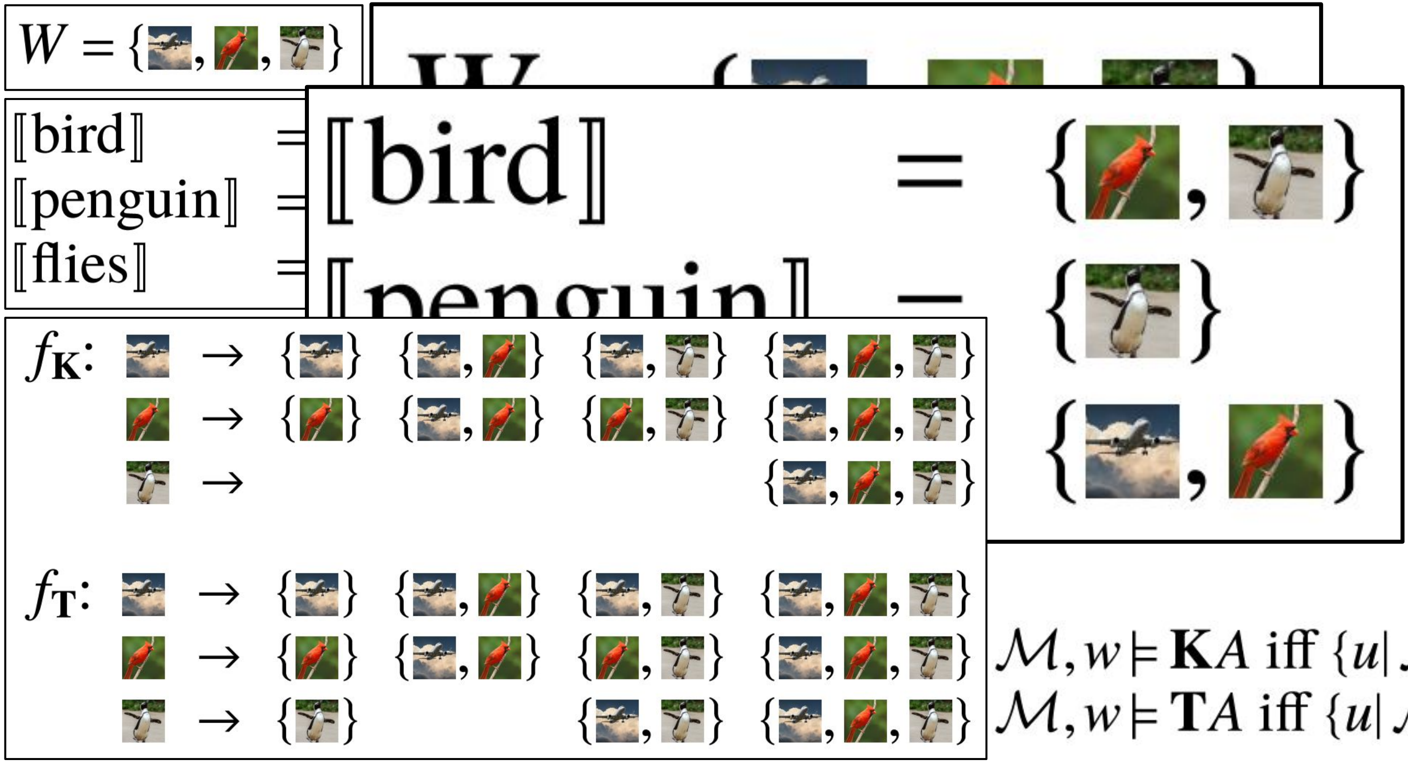
$$\frac{A \qquad A \to B}{B}$$

$$(A \to B) \wedge (B \to C) \to (A \to C)$$
$$(\neg A \to \bot) \to A$$
$$(A \wedge \neg A) \to B$$

$$\mathbf{K}A \to A$$
$$\mathbf{K}A \to \mathbf{KK}A$$
$$\mathbf{K}(A \to B) \to (\mathbf{K}A \to \mathbf{K}B)$$

$$\mathbf{T}A \to A$$
$$\mathbf{T}A \to \mathbf{TT}A$$

(Modal) Logic Models



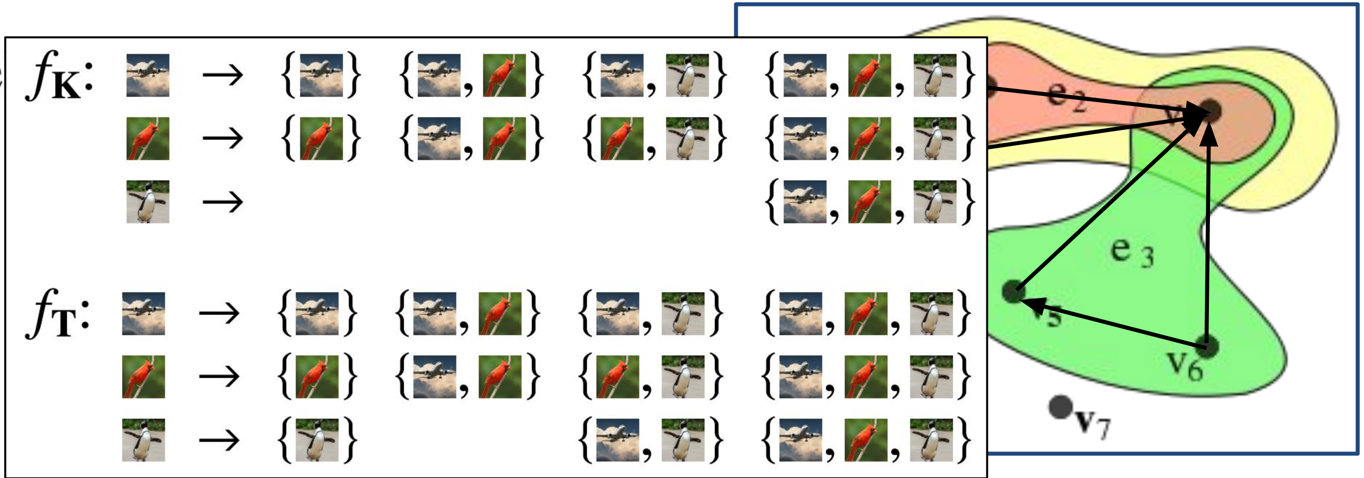
$$\mathcal{M}, w \models \mathbf{K}A \text{ iff } \{u \mid \mathcal{M}, u \models A\} \in f_{\mathbf{K}}(w)$$
$$\mathcal{M}, w \models \mathbf{T}A \text{ iff } \{u \mid \mathcal{M}, u \models A\} \in f_{\mathbf{T}}(w)$$

Eric Pacuit. *Neighborhood semantics for modal logic.*

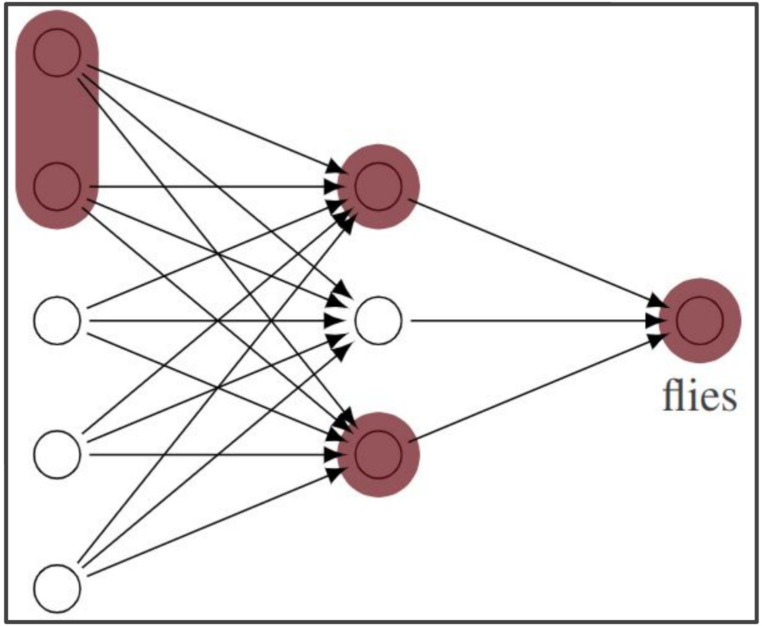
(Modal) Logic Models

$$\mathcal{M} = \langle W, f_{\mathbf{K}}, f_{\mathbf{T}}, [\![p]\!] \rangle$$

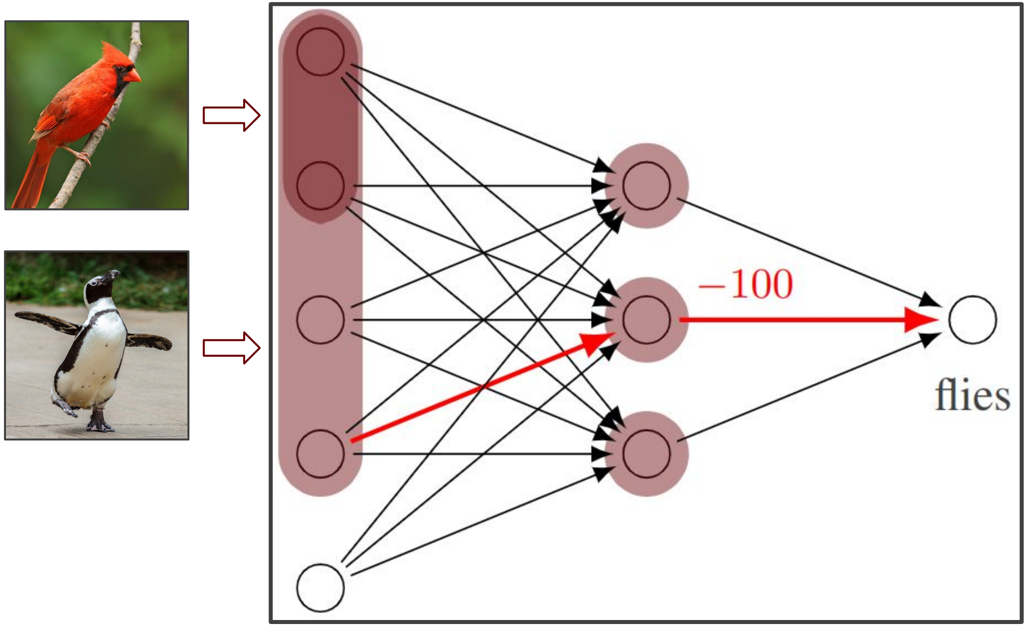
- $f_{\mathbf{K}}$ is reflexive, transitive, acyclic, monotonic
- $f_{\mathbf{T}}$ is reflexive, transitive, **not** monotonic
- $f_{\mathbf{K}}$ is a ske



Artificial Neural Networks



Artificial Neural Networks



Artificial Neural Networks

$$\mathcal{N} = \langle N, E, W, A \rangle$$

- E is feed-forward (no cycles)
- A is monotonically increasing
- A is binary

Neuro-Symbolic AI

Sebastian Bader and Pascal Hitzler. Dimensions of neural-symbolic in
Artur S d'Avila Garcez, Luis C Lamb, and Dov M Gabbay. *Neural-symbolic cognitive reasoning*.

A Semantic Theory

$$p \quad \bot \quad \neg A \quad A \to B \quad \mathbf{K}A \quad \mathbf{T}A$$

$$\begin{aligned} [\![p]\!] &= \text{some } S_p \text{ in Set} \\ [\![\neg A]\!] &= [\![A]\!]^{\mathsf{C}} \\ [\![A \to B]\!] &= \text{“}[\![A]\!] \supseteq [\![B]\!]\text{”} \end{aligned}$$

$$[\![\mathbf{K}A]\!] = \mathrm{op}([\![A]\!])$$

*Set $= \mathcal{P}(N)$

*Officially, $[\![A \to B]\!] = [\![A]\!]^{\mathsf{C}} \cap [\![B]\!]$

Hannes Leitgeb. Nonmonotonic reasoning by inhibition nets.
Hannes Leitgeb. Nonmonotonic reasoning by inhibition nets II.
Simon Odense and Artur d'Avila Garcez. A semantic framework for neural-symbolic computing.

Types of Closure Operators on Nets

$\text{Reach} : \text{Set} \to \text{Set}$

$\text{Reach}(S) = \text{The set of neurons graph-reachable from } S$

$$\boxed{[\![\mathbf{K}A]\!] = \text{Reach}([\![A]\!])}$$

$\text{Reach}^{\downarrow} : \text{Set} \to \text{Set}$

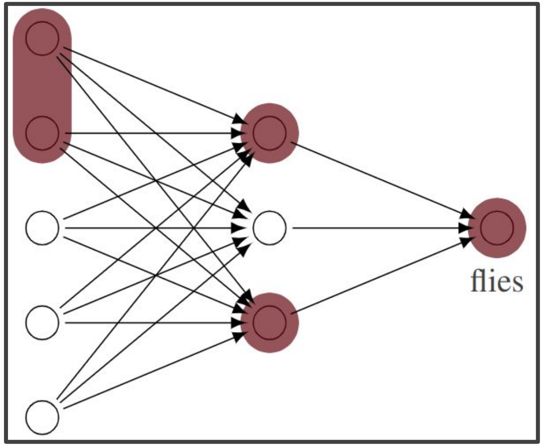
$\text{Reach}^{\downarrow}(S) = \text{The set of neurons that graph-reach some } n \text{ in } S$

$$\boxed{[\![\mathbf{K}^{\downarrow}A]\!] = \text{Reach}^{\downarrow}([\![A]\!])}$$

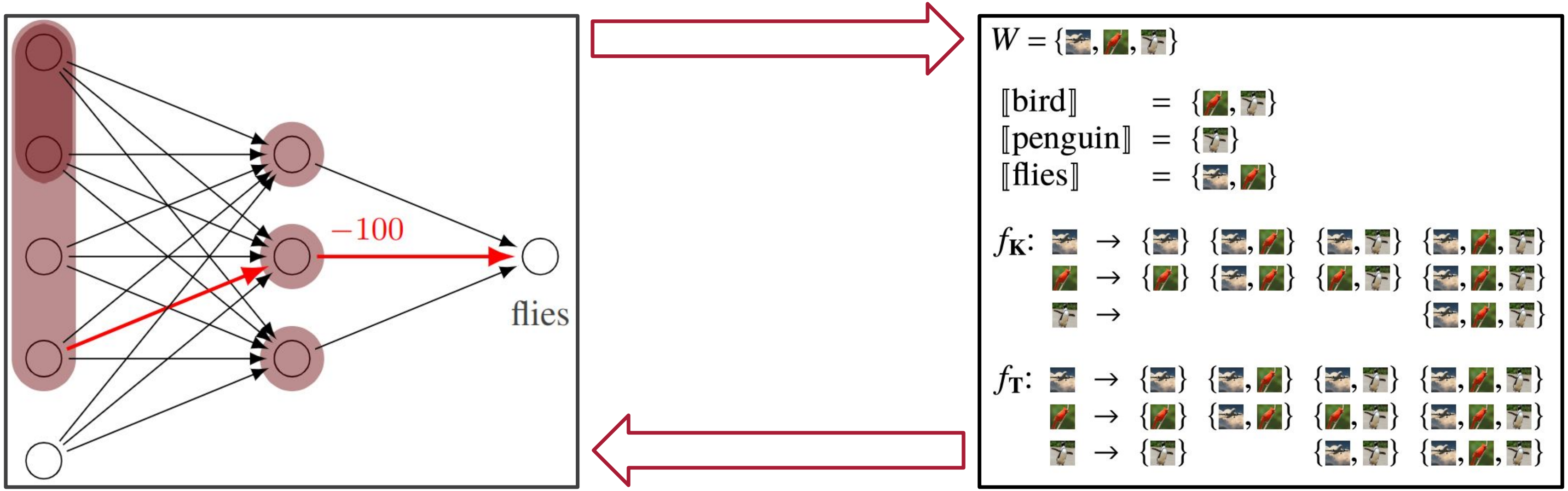
$\text{Prop} : \text{Set} \to \text{Set}$

$\text{Prop}(N) = \text{The set of neurons activated by } S$

$$\boxed{[\![\mathbf{T}A]\!] = \text{Prop}([\![A]\!])}$$



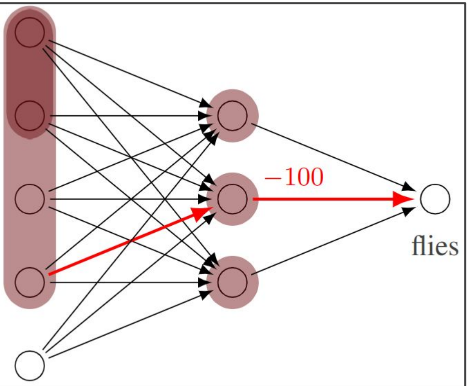
From Neural Networks to Models and Back

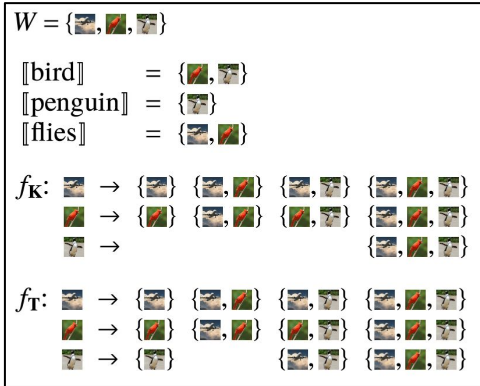


Building Models from Nets

$$\langle N, E, W, A, [\![p]\!]_{\mathcal{N}} \rangle \longrightarrow \langle W, f_{\mathbf{K}}, f_{\mathbf{T}}, [\![p]\!]_{\mathcal{M}} \rangle$$

$$\boxed{\begin{aligned} & W = N \\ & f_{\mathbf{K}}(w) = \{S \subseteq W \mid w \in \mathsf{Reach}(S)\} \\ & f_{\mathbf{T}}(w) = \{S \subseteq W \mid w \in \mathsf{Prop}(S)\} \\ & [\![p]\!]_{\mathcal{M}} = [\![p]\!]_{\mathcal{N}} \end{aligned}}$$

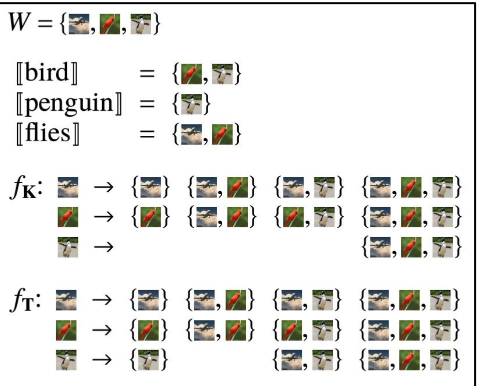


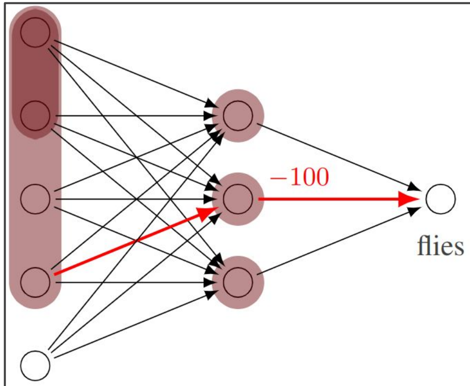


Building Nets from Models

$$\langle W, f_{\mathbf{K}}, f_{\mathbf{T}}, [\![p]\!]_{\mathcal{M}} \rangle \longrightarrow \langle N, E, W, A, [\![p]\!]_{\mathcal{N}} \rangle$$

$$N = W$$
$$(m_i, n) \in E \text{ iff } n \in \bigcap_{X \in f(m)} X$$
$$W(m_i, n) = \text{arbitrary}$$
$$A^{(n)}(\vec{x}, \vec{w}) = 1 \text{ iff } \{m_i | x_i = 1\} \in g(n)$$
$$[\![p]\!]_{\mathcal{N}} = [\![p]\!]_{\mathcal{M}}$$





Neural Network Axioms

$$\mathbf{K}(A \to B) \to (\mathbf{K}A \to \mathbf{K}B)$$
$$\mathbf{K}A \to A$$
$$\mathbf{K}A \to \mathbf{K}\mathbf{K}A$$
$$\mathbf{K}(\mathbf{K}(A \to \mathbf{K}A) \to A) \to A$$

$$\mathbf{K}^{\downarrow}(A \to B) \to (\mathbf{K}^{\downarrow}A \to \mathbf{K}^{\downarrow}B)$$
$$A \to \mathbf{K}\langle \mathbf{K}^{\downarrow} \rangle A$$
$$A \to \mathbf{K}^{\downarrow}\langle \mathbf{K} \rangle A$$

$$\mathbf{T}A \to A$$
$$\mathbf{T}A \to \mathbf{T}\mathbf{T}A$$

$$\boxed{(\mathbf{T}A \to \mathbf{K}^{\downarrow}B) \leftrightarrow (\mathbf{T}(\mathbf{T}A \vee \mathbf{K}^{\downarrow}B) \to \mathbf{K}^{\downarrow}B)}$$

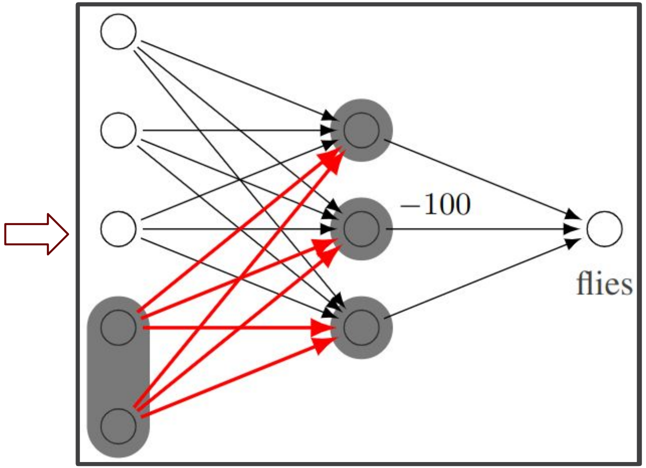
What About…

- Real-valued neuron activation?
    - Lifting binary logic to fuzzy logic
      (TODO, but shouldn't be hard)
- Learning?
    - *Modal Logic natively supports update!*
      (See next slide!)

Laura Giordano, Valentina Gliozzi, and Daniele Theseider DuprÉ. A conditional, a fuzzy and a probabilistic interpretation of self-organizing maps. *Journal of Logic and Computation*, 32(2):178–205, 2022.
Caleb Kisby, Saúl Blanco, and Lawrence Moss. The logic of hebbian learning.

(Naïve) Hebbian Learning

Neurons that fire together wire together



$$\Delta W_{ij} = \eta x_i x_j$$

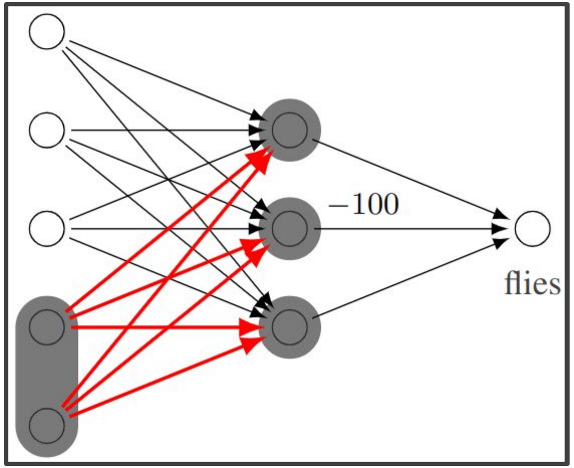
Donald Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

Hebbian Learning as a Closure Operator

$\mathsf{Inc}^* : \mathsf{Net} \times \mathsf{Set} \rightarrow \mathsf{Set}$

$\mathsf{Inc}^*(\mathcal{N}, S) =$ The net obtained by **maximally** strengthening all weights within $\mathsf{Prop}(S)$

$$\boxed{[\![[A]^* B]\!]_{\mathcal{N}} = [\![B]\!]_{\mathsf{Inc}(\mathcal{N}, [\![A]\!])}}$$



$$\Delta W_{ij} = \eta x_i x_j$$

The Hebbian Learning Axioms

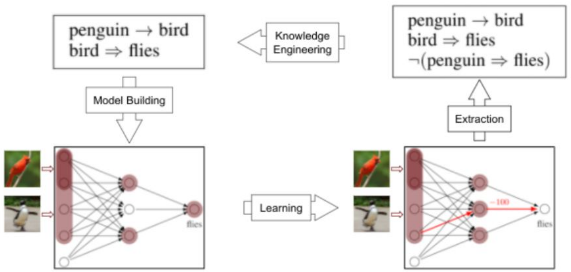
$$
\begin{array}{lcl}
[A]^*p & \leftrightarrow & p \\
[A]^*\neg B & \leftrightarrow & \neg[A]^*B \\
[A]^*(B \to C) & \leftrightarrow & [A]^*B \to [A]^*C \\
[A]^*\mathbf{K}B & \leftrightarrow & \mathbf{K}[A]^*B \\
 & & \\
[A]^*\mathbf{T}B & \leftrightarrow & [(\mathbf{T}A \vee \mathbf{T}B \leftrightarrow \bot) \to \mathbf{T}[A]^*B] \\
 & & \vee \quad [\neg(\mathbf{T}A \vee \mathbf{T}B \leftrightarrow \bot) \to \mathbf{T}[A]^*B \wedge (\mathbf{T}A \vee \mathbf{K}B)]
\end{array}
$$

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

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

README.md

à la Mode: A Neural Network Model Builder



A Python (Tensorflow) suite for neural network **checking** ("does the net satisfy P?") and neural network **building** ("construct a net that satisfies P.") We do this by leveraging a neuro-symbolic translation between logical symbols and sets of neurons in the network.

à la Mode is the more hasty and confident sister of Argyle. Both share the same core features, but à la Mode releases faster and is easier to use (for those Python+Tensorflow inclined). However, unlike Argyle, à la Mode is not formally verified (although it's models are proved correct by hand).

NOTE: This program is currently **very much in development**, and many of the planned features involve significant research efforts (this is my PhD). So what the program can do right now is somewhat limited.

3 watching

0 forks

Releases

1 tags

Create a new release

Packages

No packages published

Publish your first package

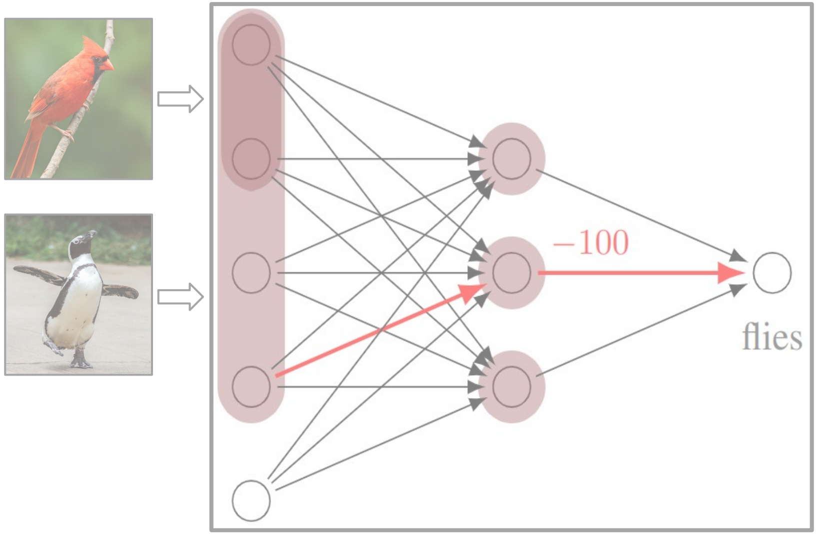
Languages

Python 85.8% **Tcl** 14.2%

Future Work:

1. Recurrent neural networks
2. First-order and higher-order logics
3. Other learning operators (e.g. stable Hebbian, backpropagation)

Questions?



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