Qualifying Exam Defense

Caleb Kisby

The Neuro-Symbolic Schism

Knowledge-Based Paradigm

- Considerable success with small, richly structured domains
- A little knowledge goes a long way
- A compelling psychological model

but...

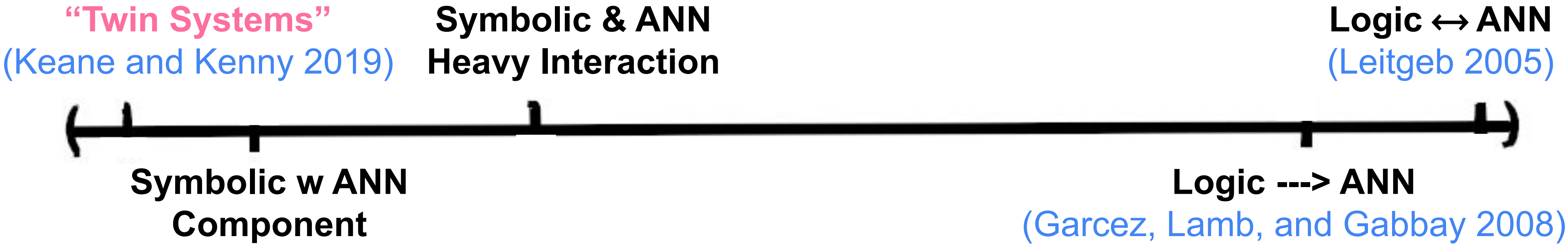
- No or very conservative learning
- Knowledge-engineering is hard!

Connectionist Paradigm

- Wildly successful at learning
- *Very little* knowledge also goes a long way
- A compelling neural model

but...

- Requires a ton of data to learn
- Unclear what an ANN believes
  - Unexplainable decisions



Weakly-Coupled **Strongly-Coupled**

The Spectrum of Neuro-Symbolic Proposals

Response 1:
Playing Notakto

Question 1 – Playing Notakto

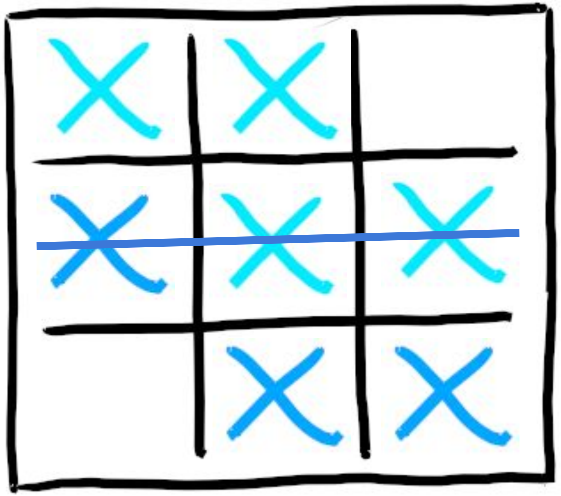
Consider the following two-player game on an n × n board, with $n \leq 6$. The two players take turns filling the board with the same symbol "X." The first player that completes a row, a column, or the main diagonal and anti-diagonal loses the game.

Expectations: For this question, the following is expected.

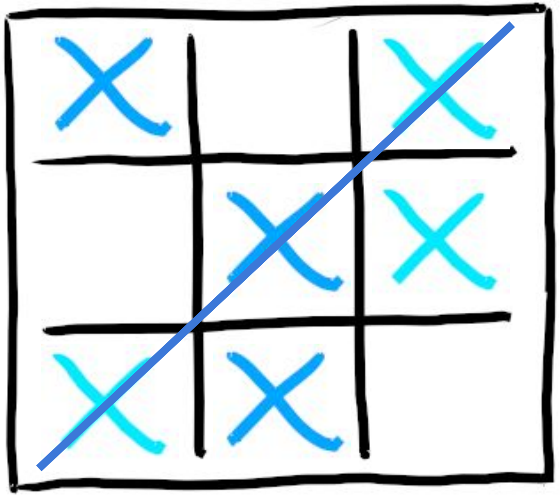
a) Implement a neural network architecture to play the game. To aid you in your task, you are allowed to research network architectures that have been used for other games and you might use common open-source libraries like tensor flow.
b) You should try different architectures and parameters, and eventually decide on specific choices.
c) You need to write a short (around 3 to 5 pages) report that addresses the choice of architecture, parameters, and how you asses the performance of your network.
d) The report should also include observations you find interesting during the training process. In particular, if the training process provides any insight that can be translated into strategies to play the game.

The Game of Notakto

(Chow 2010)
(Plambeck and Whitehead 2013)



Player 2 Wins!



Player 1 Wins!

a	b	c	d
e	f	g	h
b	a	d	c
f	e	h	g

How to Win at Notakto

n	Winner under perfect play	Reference
1	2nd Player	
2	1st Player	
3	1st Player	
4	2nd Player	(Chow 2010)
5	1st Player	(Chow 2010)
6	1st Player	(Chen et al. 2021)
$n = 4k$	2nd Player	(Chen et al. 2021)

AlphaGo Zero to the Rescue?

AlphaGo Zero (Silver et al. 2017)

- *Generic* ANN-Based framework for board games
- Chess, Go, …
- Converges on a winning strategy for Tic-Tac-Toe
- Why not Notakto?

(Chen et al. 2021)

- AlphaGo Zero fails to learn competent play for small board sizes $n > 5$
- Rare nature of competent Notakto boards
- MCTS and sampling method also encode prior knowledge (Marcus 2018)

“Twin Systems”
(Keane and Kenny 2019)

Symbolic & ANN
Heavy Interaction

Logic ↔ ANN
(Leitgeb 2005)

Symbolic w ANN
Component

Logic ---> ANN
(Garcez, Lamb, and Gabbay 2008)

Weakly-Coupled **Strongly-Coupled**

The Spectrum of Neuro-Symbolic Proposals

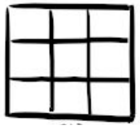
Training Boards

(Lai 2015)

Generating the Boards

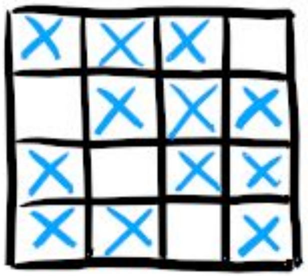
- Distribution
  - Boards should reflect competent play
- Variety
  - e.g. Boards that are unfairly stacked against player
- Volume
  - Enough boards for ANN to generalize

Evaluating the Boards

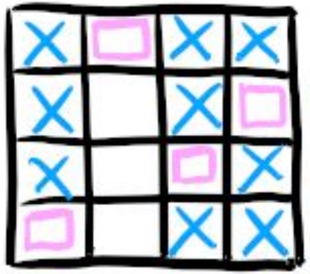


Feature Representation

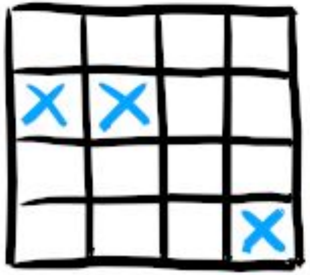
Turn Flag



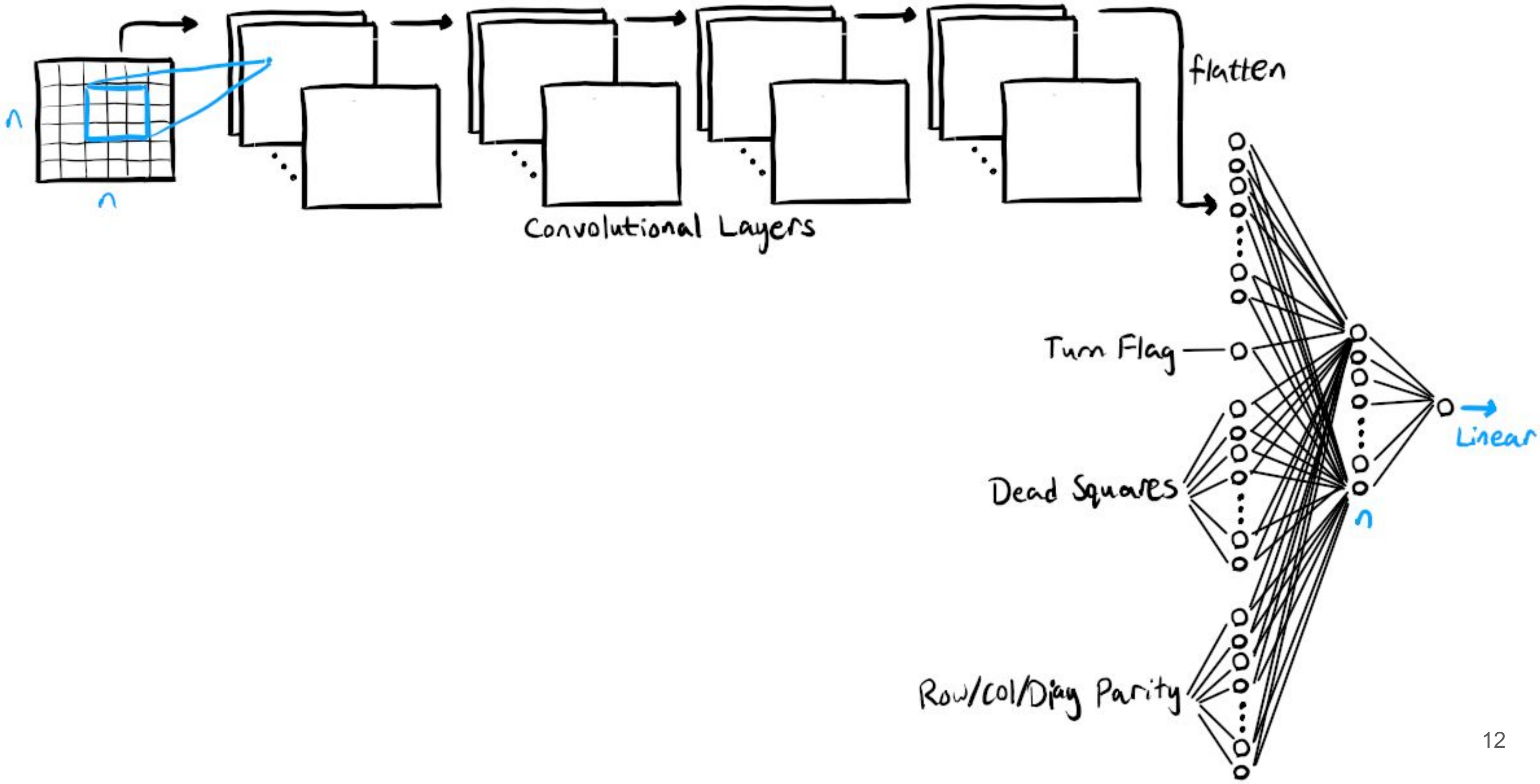
Dead Squares



Row Parity



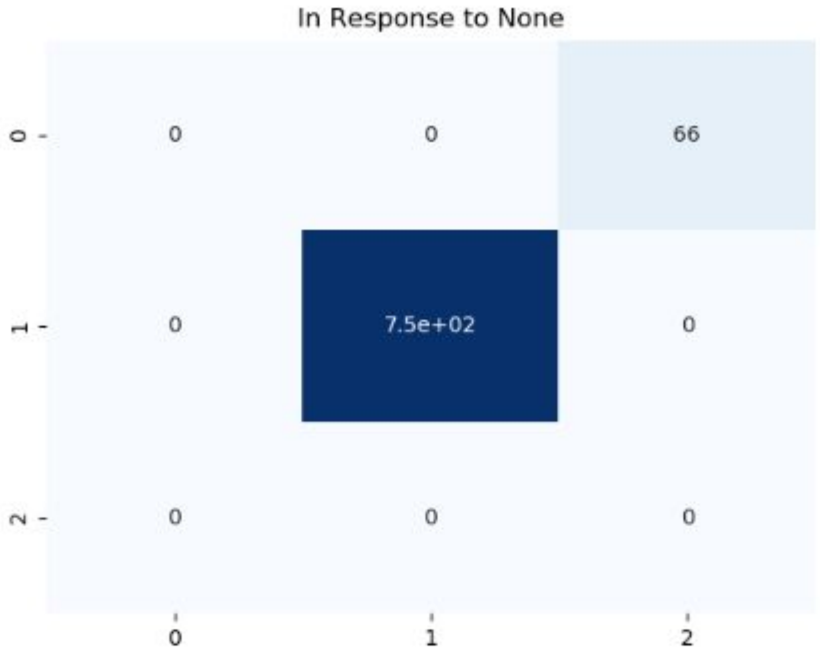
Neural Network Architecture



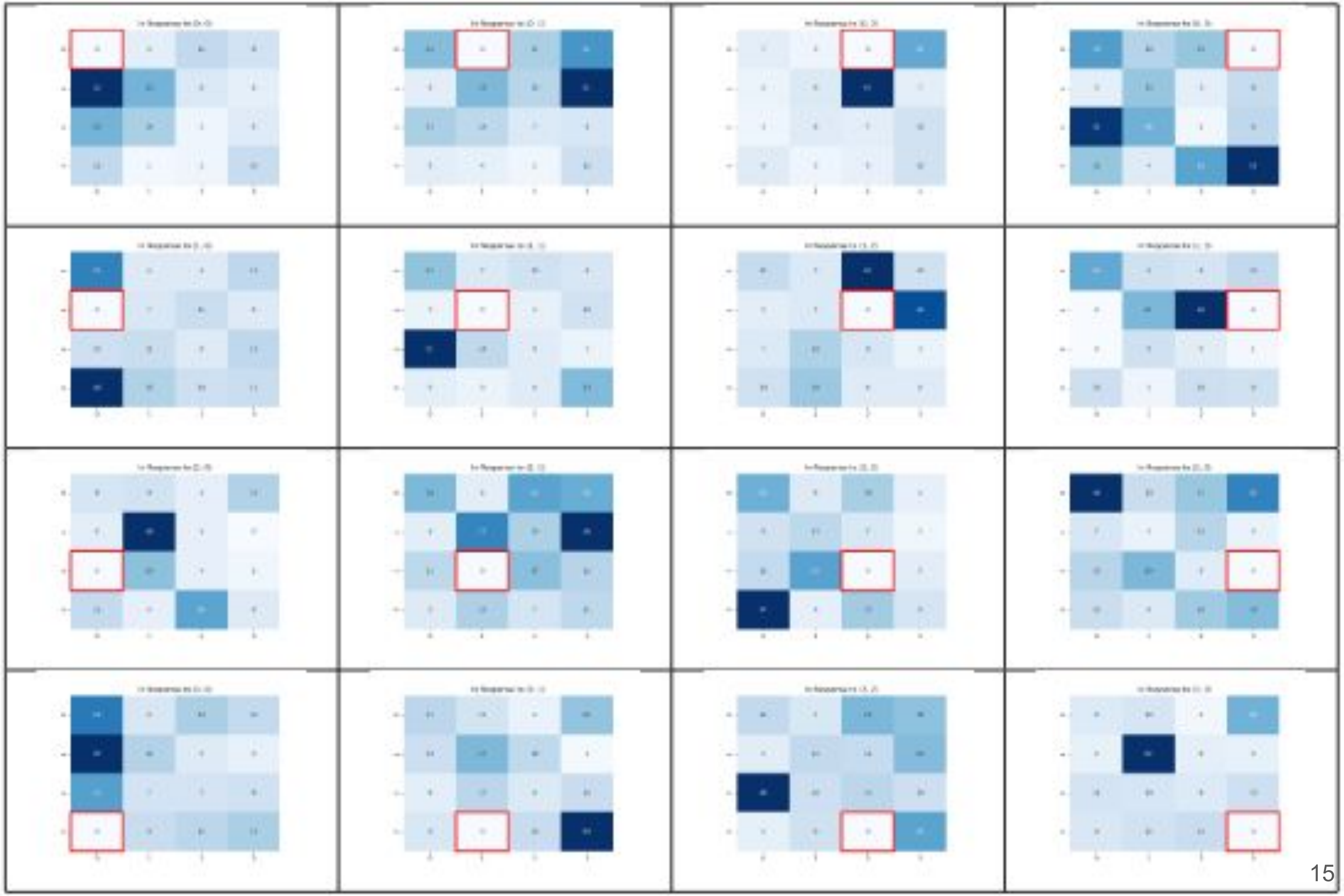
n	Opponent	Featured Net	AlphaZero Net
$n=3$	Itself	100% / 0%	92% / 8%
	Featured Net		100% / 1%
	AlphaZero Net	99% / 0%	
	Random	94% / 82%	90% / 84%
	Greedy	79% / 26%	88% / 20%
$n=4$	Itself	8% / 92%	92% / 8%
	Featured Net		18% / 82%
	AlphaZero Net	18% / 82%	
	Random	93% / 94%	91% / 96%
	Greedy	37% / 55%	49% / 54%
$n=5$	Itself	59% / 41%	41% / 59%
	Featured Net		47% / 49%
	AlphaZero Net	51% / 53%	
	Random	99% / 99%	99% / 96%
	Greedy	54% / 48%	60% / 39%
$n=6$	Itself	23% / 77%	69% / 31%
	Featured Net		33% / 53%
	AlphaZero Net	47% / 67%	
	Random	100% / 99%	99% / 98%
	Greedy	52% / 47%	55% / 59%

Mapping the Net's Strategy

- We try to extract the ANN's strategy from a heat map
- 1000 games against Greedy
- Collect net's *responses* to moves
- Only keep responses that resulted in a **win**



a	b	c	d
e	f	g	h
b	a	d	c
f	e	h	g



Using an LSTM Instead
(Near-Future Work)

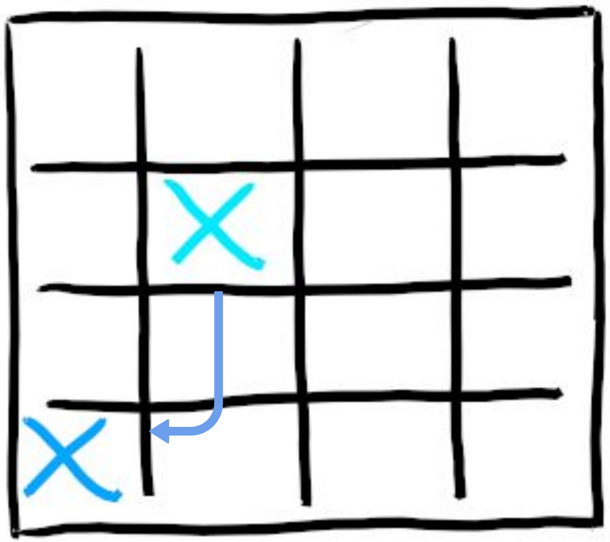
Note:

- Winning strategies depend on *what opponent played last*
- Contrast with Chess, Go, Tic-Tac-Toe, ...

Hypothesis:

- Can’t evaluate a single static board
- We need sequential information

So I should use, e.g. a ConvLSTM to learn Notakto.



Reconsidering Features
(Near-Future Work)

(Lai 2015)

“For neural networks to work well, the feature representation needs to be relatively smooth in how the input space is mapped to the output space. Positions that are close together in the feature space should, as much as possible, have similar evaluations.”

Q&A

References:

Chen, Z.; Wang, C.; Laturia, P.; Crandall, D.; and Blanco, S. 2021. **How to play Notakto: Can reinforcement learning achieve optimal play on combinatorial games?**

Chow, T. 2010. **Neutral tic tac toe.** MathOverflow. URL:https://mathoverflow.net/q/24693 (version: 2021-03-15).

Lai, M. 2015. **Giraffe: Using deep reinforcement learning to play chess**. arXiv preprint arXiv:1509.01549.

Marcus, G. 2018. **Innateness, alphazero, and artificial intelligence**. arXiv preprint arXiv:1801.05667.

Nair, S. 2018. **Alpha zero general (any game, any framework!).** https://github.com/suragnair/alpha-zerogeneral

Plambeck, T. E., and Whitehead, G. 2013. **The secrets of notakto: Winning at x-only tic-tac-toe**. arXiv preprint arXiv:1301.1672.

Silver, D. et al. 2017. **Mastering the game of go without human knowledg**e. *Nature* 550(7676):354–359.

Github:

https://github.com/ais-climber/notakto-player/tree/main/v1_feedforward_net

Response 2
Hudson & Manning's
Neural State Machine

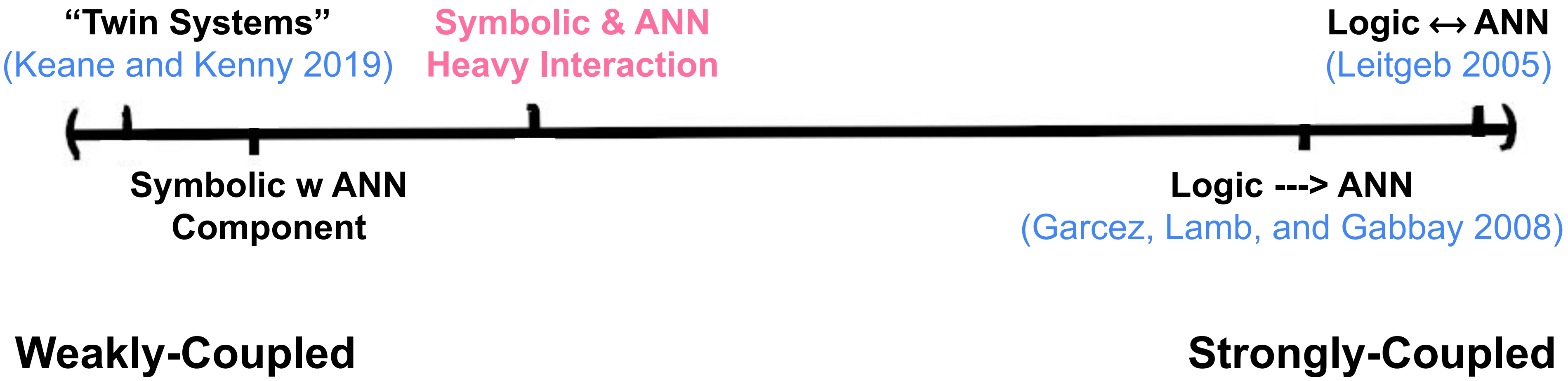
Question 2 – Response to Leitgeb, Response to Hudson & Manning

This question is intended to be an open-ended, "deep dive" into literature topics pertaining to the relation between symbolic and neural approaches in AI. To start, read two papers:

- Hannes Leitgeb's "From Logic to Neural Networks and Back", and also
- Drew Hudson and Chris Manning's paper https://arxiv.org/pdf/1907.03950.pdf proposing a "neural state machine".

Leitgeb's paper is old and there might be newer work by him and others, and so there might be newer work. Certainly there are videos of talks by Leitgeb. Hudson and Manning's paper is newer, and for this you might also want to find older papers on automata and neural nets.

Your task in this problem is to write a review paper that presents the two papers that you choose to read. What are their aims, what do they do, and what do they not do? Most importantly, you should try with each of these to "take the next step" on their lines of work. The minimum here would be to say how one would extend each paper in some interesting way, 1 or to challenge it. The maximum would of course be to carry out such an extension or challenge.



The Spectrum of Neuro-Symbolic Proposals

The Neural State Machine

- VQA Task
- Given image and question, constructs state machine
  - **States:** Objects in the scene
  - **Transitions:** Concepts and relationships
  - **Probability distribution** of states we are in
  - **Instructions** are extracted from the question

Where are ANNs Used?

- CNN for parsing image, boundary detection
- Concept vector embeddings learned in advance
- To determine what state to transition into

The system builds a reasoner using ANN submodules

Q&A

References:

Hudson, D. A., and Manning, C. D. 2019. **Learning by abstraction: The neural state machine**. arXiv preprint arXiv:1907.03950.

Response 3:
A Neuro-Symbolic Approach to Case Adaptation

Question 3 – A Neuro-Symbolic Approach to Case Adaptation

The primary locus of reasoning in case-based reasoning systems is case adaptation, generally based on declarative domain knowledge. In contrast, deep neural networks exploit a knowledge-light statistical approach. Briefly survey previous research on integrations of CBR and neural networks for case adaptation. Propose a promising way to integrate neural network and knowledge-based case adaptation, highlighting how it contrasts with prior approaches. Describe how the integration could be done, issues likely to arise, and how those issues could be addressed (or why they remain open problems).

Learning Case Adaptation

(Hanney and Keane 1997)
(Liao, Liu, and Chao 2018)
(Ye et al. 2021)

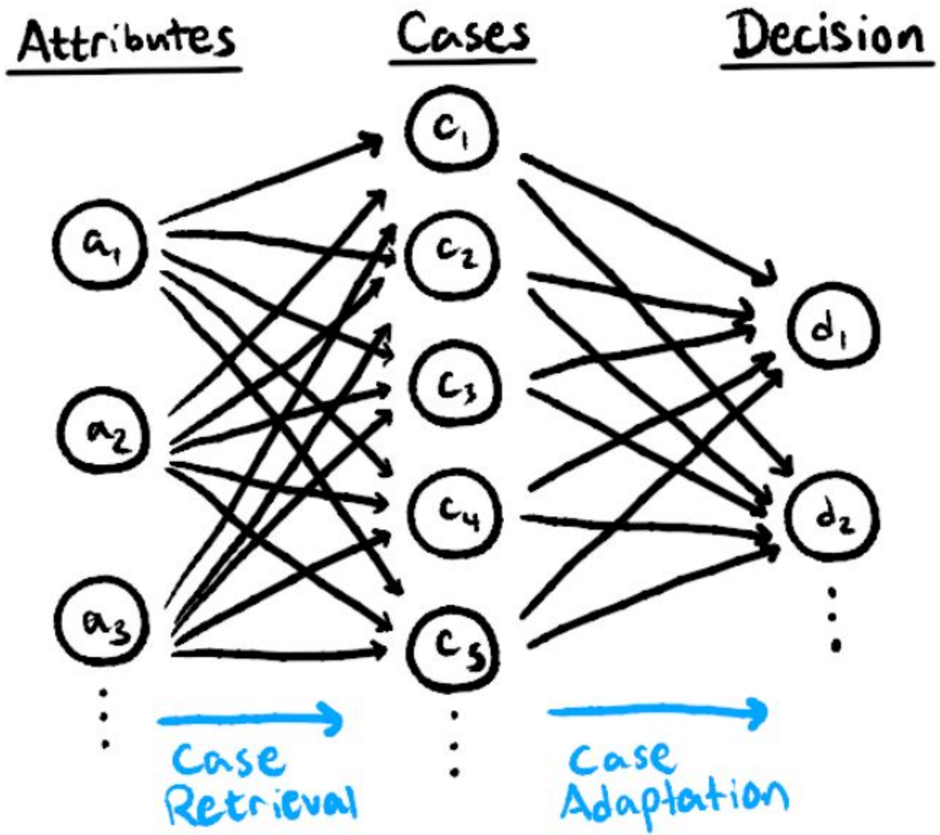
- Crafting case adaptation rules is hard!
  - Knowledge Engineering is hard!

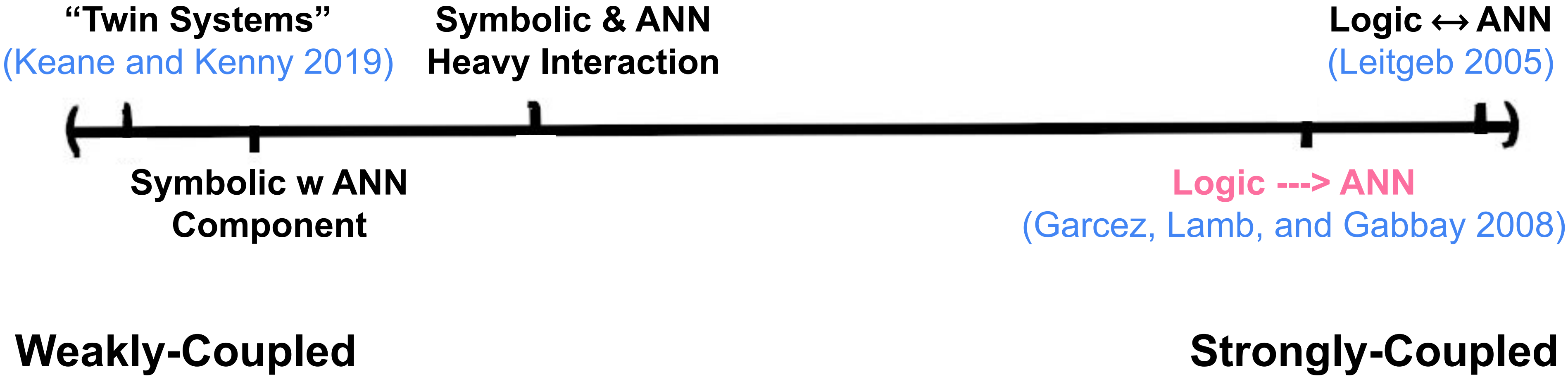
Case Difference Heuristic (CDH)

- Feed ANN *differences between cases* A, B and output for case A
- ANN predicts output for case B

Neural Models of CBR

(Becker and Jazayeri 1989)





The Spectrum of Neuro-Symbolic Proposals

The CILP Translation

(Garcez, Lamb, and Gabbay 2008)

- General logic program → neural network
- I'll illustrate via *propositional adaptation rules*
- The real result:
    - Modal logic
    - Temporal logic
    - (Some) predicate logic

CILP In Action
A Toy CHEF Adaptation

- The classic CHEF system
- I consider a much smaller domain:
  - Cases: *Sandwich recipes, with some odd tastes*
  - Ingredients: *Bread, cheese, chili peppers, sriracha sauce*
  - Task: *Determine if a sandwich is going to be spicy*

Given sandwiches A and B (say we know A is spicy):

If B has a spicy ingredient, then B is spicy…?

If B has a spicy ingredient and no neutralizing ingredient…?

If B has a spicy ingredient, and has a neutralizing ingredient ***only if*** *A does.*

CILP In Action
In Propositional Form

b_A : A has bread
c_A : A has cheese
s_A : A has sriracha sauce
p_A : A has chili peppers
H_A : A is spicy

b_B : B has bread
c_B : B has cheese
s_B : B has sriracha sauce
p_B : B has chili peppers
H_B : B is spicy

$$H_A \wedge (\sim b_B \vee b_A) \wedge (\sim c_B \vee c_A) \wedge (s_B \vee p_B) \rightarrow H_B$$

CILP In Action

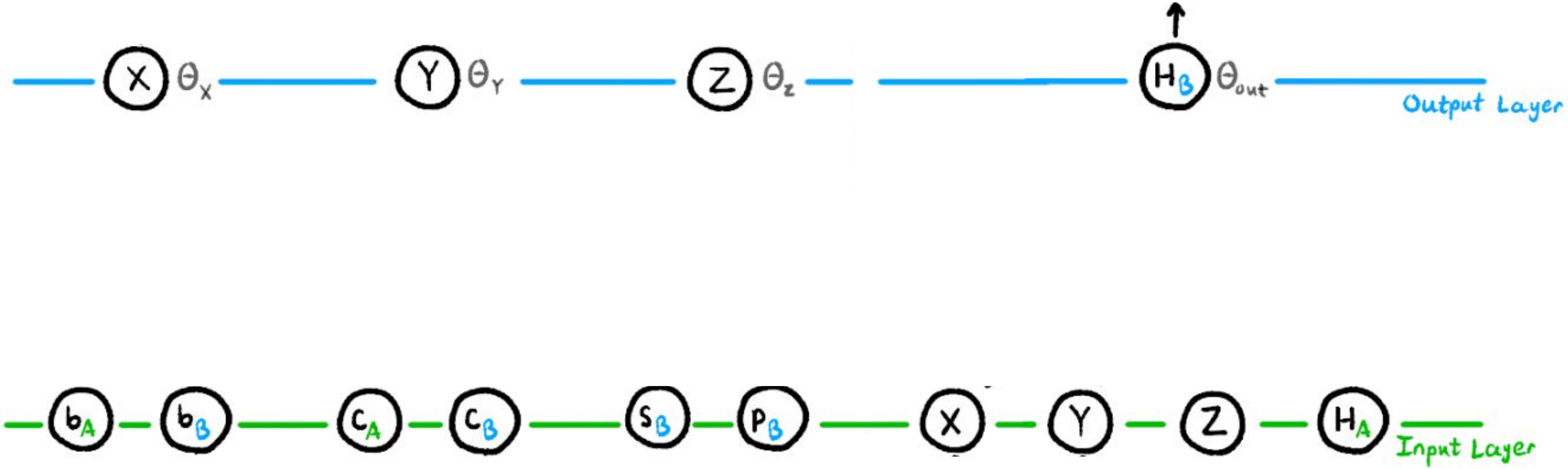
In General Logic Programming Form

$$H_A \wedge (\sim b_B \vee b_A) \wedge (\sim c_B \vee c_A) \wedge (s_B \vee p_B) \rightarrow H_B$$

$$\{ \quad \}$$

CILP In Action
In Neural Network Form

$$\{\sim b_B \rightarrow X; b_A \rightarrow X;\\ \sim c_B \rightarrow Y; c_A \rightarrow Y;\\ s_B \rightarrow Z; p_B \rightarrow Z;\\ X, Y, Z, H_B \rightarrow H_A\}$$



What This Proposal is Good For

Problems Solved:

- We can encode *known* adaptation rules in a neural network
  - Even if rules are partial or unsound
- Network then *refines* our adaptation rules

Problems Left Open:

- How can we back-convert to get new adaptation rules after learning?
- How do we know the net is learning adaptation?

A ‘Refined’ CHEF
(Near-Future Work)

- I only showed the propositional case
- (Garcez, Lamb, Gabbay 2008) extend CILP for:
  - Modal Logic
  - Temporal Logic
  - Predicate Logic

My Plan:

- micro-CHEF adaptation
  - Uses rules with predicates
  - Tricky part: Handling ‘recipe critic’

Q&A

References:

Becker, L., and Jazayeri, K. 1989. **A connectionist approach to case-based reasoning**. In *Proceedings of the Case-Based Reasoning Workshop*, 213–217. Morgan Kaufmann San Mateo, CA.

Garcez, A. S.; Lamb, L. C.; and Gabbay, D. M. 2008. **Neural-symbolic cognitive reasoning**. *Springer Science & Business Media*.

Hanney, K., and Keane, M. T. 1997. **The adaptation knowledge bottleneck: How to ease it by learning from cases**. In *International Conference on Case-Based Reasoning*, 359–370. Springer.

Keane, M. T., and Kenny, E. M. 2019. **How case-based reasoning explains neural networks: A theoretical analysis of xai using post-hoc explanation-by example from a survey of ann-cbr twin-systems**. In *International Conference on Case-Based Reasoning*, 155–171.Springer.

Liao, C.-K.; Liu, A.; and Chao, Y.-S. 2018. **A machine learning approach to case adaptation.** In *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 106–109. IEEE.

Ye, X.; Leake, D.; Jalali, V.; and Crandall, D. 2021a. **Learning adaptations for case-based classification: A neural network approach**.

Ye, X.; Zhao, Z.; Leake, D.; Wang, X.; and Crandall, D. 2021b. **Applying the case difference heuristic to learn adaptations from deep network features**. arXiv preprint arXiv:2107.07095.

Response 2:

Neural Network Models of Logics

Question 2 – Response to Leitgeb, Response to Hudson & Manning

This question is intended to be an open-ended, "deep dive" into literature topics pertaining to the relation between symbolic and neural approaches in AI. To start, read two papers:

- Hannes Leitgeb's "From Logic to Neural Networks and Back", and also
- Drew Hudson and Chris Manning's paper https://arxiv.org/pdf/1907.03950.pdf proposing a "neural state machine".

Leitgeb's paper is old and there might be newer work by him and others, and so there might be newer work. Certainly there are videos of talks by Leitgeb. Hudson and Manning's paper is newer, and for this you might also want to find older papers on automata and neural nets.

Your task in this problem is to write a review paper that presents the two papers that you choose to read. What are their aims, what do they do, and what do they not do? Most importantly, you should try with each of these to "take the next step" on their lines of work. The minimum here would be to say how one would extend each paper in some interesting way, 1 or to challenge it. The maximum would of course be to carry out such an extension or challenge.

“Twin Systems”
(Keane and Kenny 2019)

Symbolic & ANN
Heavy Interaction

Logic ↔ ANN
(Leitgeb 2005)

Symbolic w ANN
Component

Logic ---> ANN
(Garcez, Lamb, and Gabbay 2008)

Weakly-Coupled

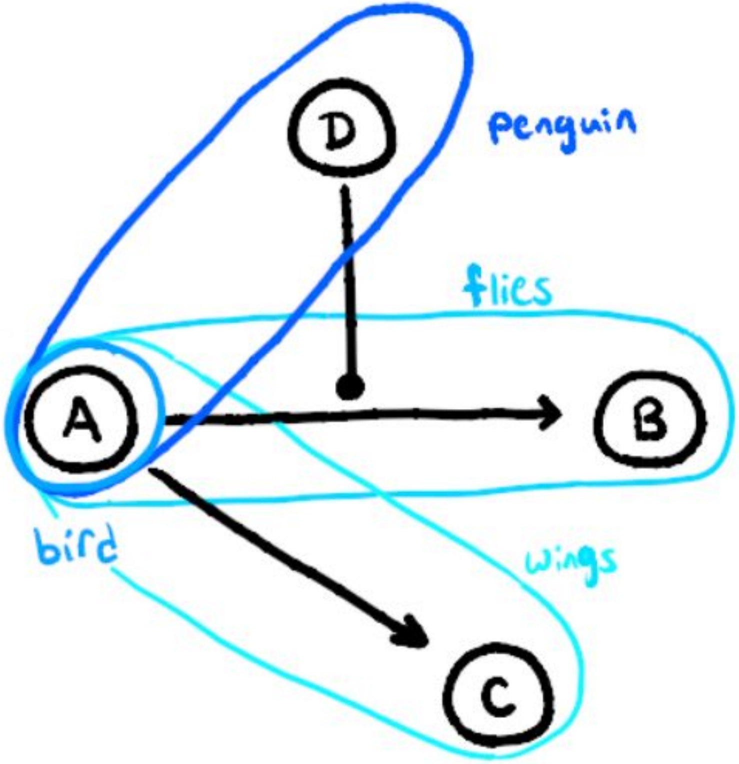
Strongly-Coupled

The Spectrum of Neuro-Symbolic Proposals

History of Formal Correspondences

- (McCulloch and Pitts 1943)
  - Inhibitory, Feedforward ↔ Propositional Logic
- (Balkenius and Gärdenfors 1991)
  - Generic RNN, Inhibitory ↔ ‘Cumulative’ Conditional Logic
  - Explored via computer simulations
- (Siegelmann and Sontag 1995)
  - Elman-RNN, weights $\in \mathbb{Q}$, *infinite precision, unbounded time* ↔ Turing
  - weights $\in \mathbb{R}$ ↔ Turing + HALT
- (Blutner 2004)
  - Hopfield Networks ↔ Poole’s Default Logic (annotated w weights)
- (Weiss, Goldberg, Yahav 2018)
  - Elman-RNN, *finite precision, bounded time* ↔ State Machines
  - LSTMs, *finite precision, bounded time* ↔ k-counter Machines (✓ a^nb^n ✓ $a^nb^nc^n$)
- **(Leitgeb 2004, 2005, 2018)**
  - **Balkenius and Gärdenfors, but…**
  - **Formal semantics & completeness proof**
  - **Restriction: Layered, Feedforward ↔ Conditionals + LOOP**

Leitgeb's Semantics for Conditionals



$[\![bird]\!] = \{A\}$ $[\![flies]\!] = \{A, B\}$
$[\![wings]\!] = \{A, C\}$ $[\![penguin]\!] = \{A, D\}$

What About Learning?

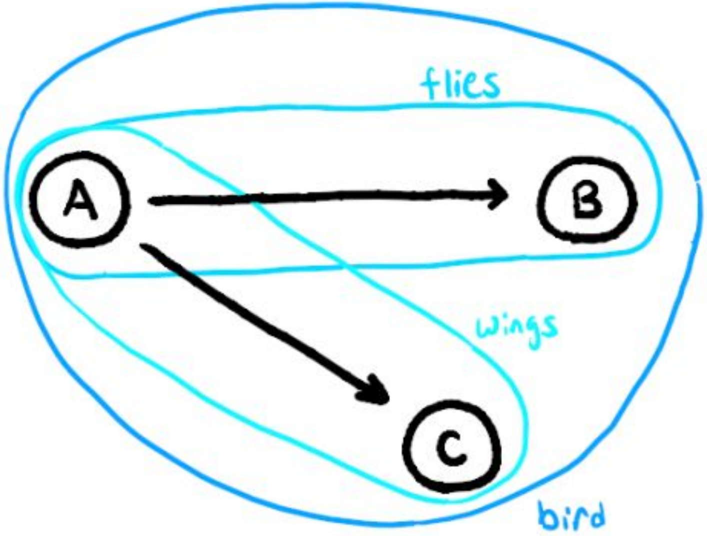
- (McCulloch and Pitts 194?)
  - Inhibit... Feedforward ... opositional Logic
- (Balkeni... ...fors 1991)
  - Generic R... ...ulative Logic C...
  - Explor... ...omputa...
- (Sieg... ...n and Sontag 199...)
  - ...man-RNN, weights ∈ ℚ, *infin...* ... Turing
  - weights ∈ ℝ ↔ Turing + P...
- (Blutner 2004)
  - Hopfield Networ... ...Logic (annotated w weights)
- (Weiss, Gold... ...8)
  - Elman... ...*, bounded time* ↔ State Machines
  - LST... ...*, bounded time* ↔ k-counter Machines (✓ a^nb^n ✓ $a^nb^nc^n$)
- **(Leitgeb ...05, 2018)**
  - **Balken... and Gärdenfors, but…**
  - **Formal semantics & completeness proof**
  - **Restriction: Layered, Feedforward ↔ C + LOOP**

The structure of th... does not change with time.

network. In the sequel, the learn... will play no significant role.

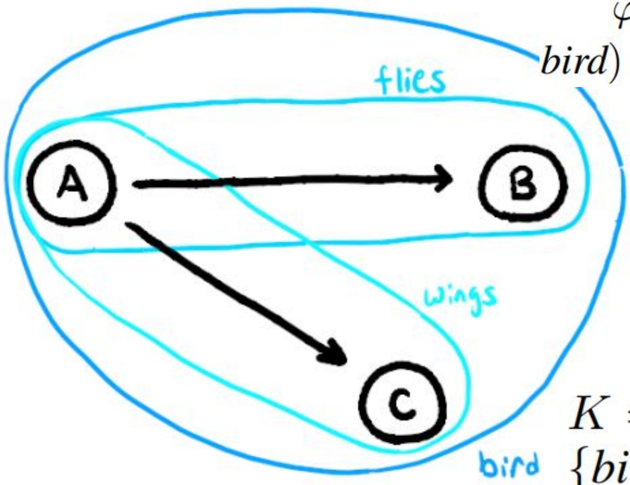
means of conditionals. However, it is as yet unknown how learning processes in networks – by which the weights in a network change under the influence of a learning algorithm and training data – can be represented by logical rules. Learning

Update via Inhibitory Signals

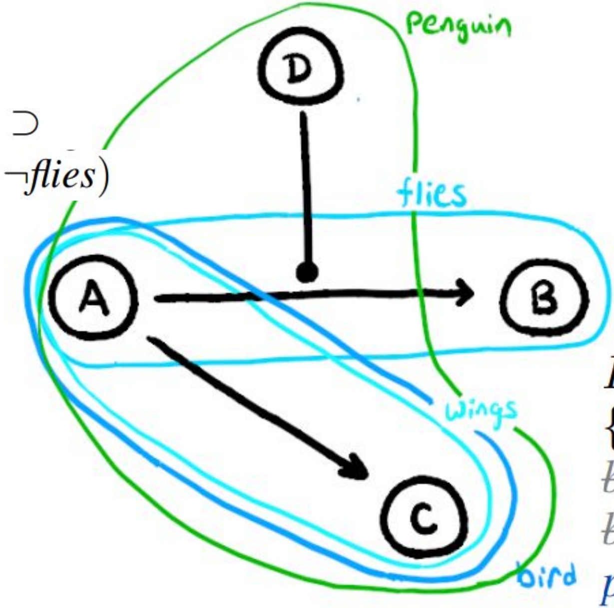


Inhibitory Update as Belief Revision
(Contraction + Expansion)

$\varphi = (\mathit{penguin} \supset \mathit{bird}) \land (\mathit{penguin} \supset \neg \mathit{flies})$



$$K = \{\mathit{bird} \supset \mathit{wings} \\ \mathit{bird} \supset \mathit{flies} \\ \mathit{bird} \supset \mathit{wings} \land \mathit{flies} \\ \ldots\}$$



$$K * \varphi = \{\mathit{bird} \supset \mathit{wings} \\ \cancel{\mathit{bird} \supset \mathit{flies}} \\ \cancel{\mathit{bird} \supset \mathit{wings} \land \mathit{flies}} \\ \mathit{penguin} \supset \mathit{bird} \\ \mathit{penguin} \supset \neg \mathit{flies} \\ \mathit{bird} \Rightarrow \mathit{flies} \\ \mathit{wings} \supset \mathit{bird} \\ \ldots\}$$

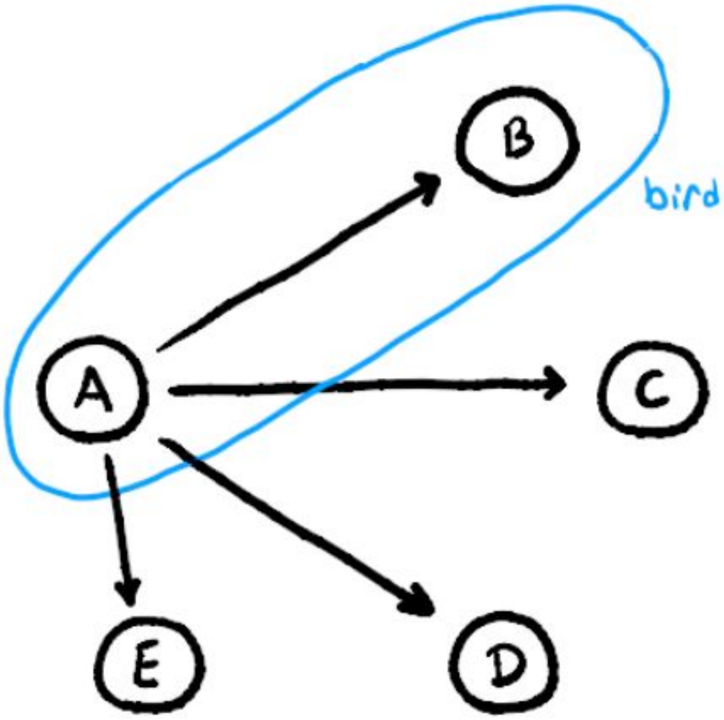
Inhibitory Update as Belief Revision
(Gärdenfors' Postulates)

$$K = \{ bird \supset wings,\ bird \supset flies,\ bird \supset wings \wedge flies,\ \ldots \} \quad \rightarrow \quad K * \varphi = \{ bird \supset wings,\ \cancel{bird \supset flies},\ \cancel{bird \supset wings \wedge flies},\ penguin \supset bird,\ penguin \supset \neg flies,\ bird \Rightarrow flies,\ wings \supset bird,\ \ldots \}$$

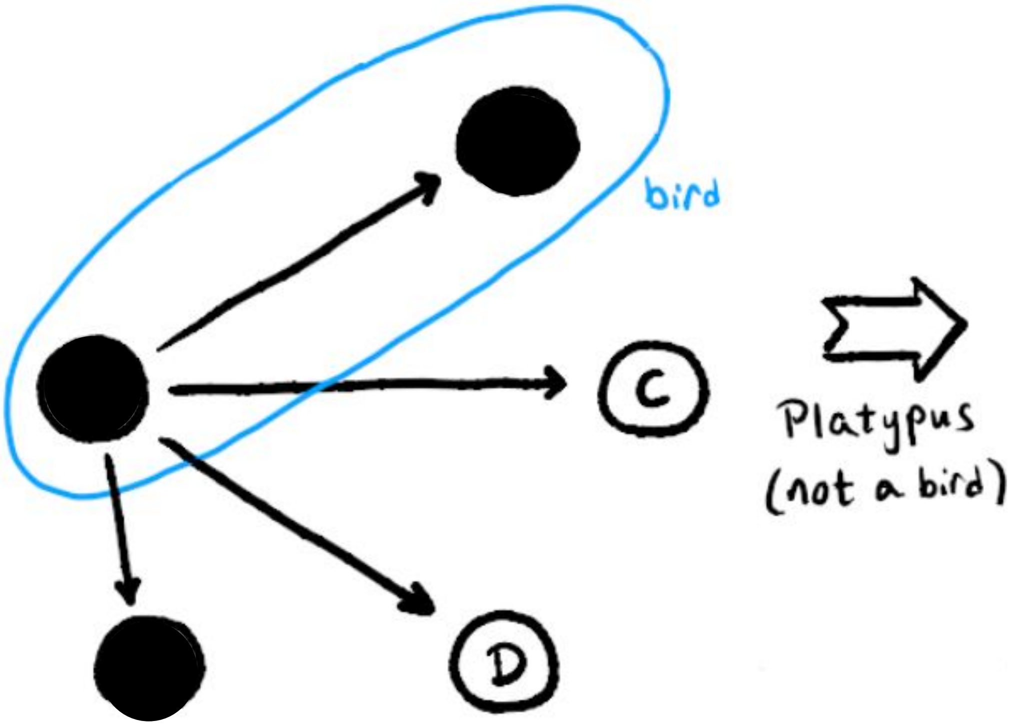
- ✓ (*Closure*): $K * \varphi = Cl(K * \varphi)$
- ✓ (*Success*): $\varphi \in K * \varphi$
- ✗ (*Inclusion*): $K * \varphi \subseteq K + \varphi$
- ✗ (*Vacuity*): If $\neg\varphi \notin K$, then $K * \varphi = K + \varphi$
- ✓ (*Consistency*): $K * \varphi$ is consistent if φ is consistent
- ✓ (*Extensionality*): If $\varphi \leftrightarrow \psi \in Cl(\emptyset)$, then $K * \varphi = K * \psi$

Update via Associative Learning

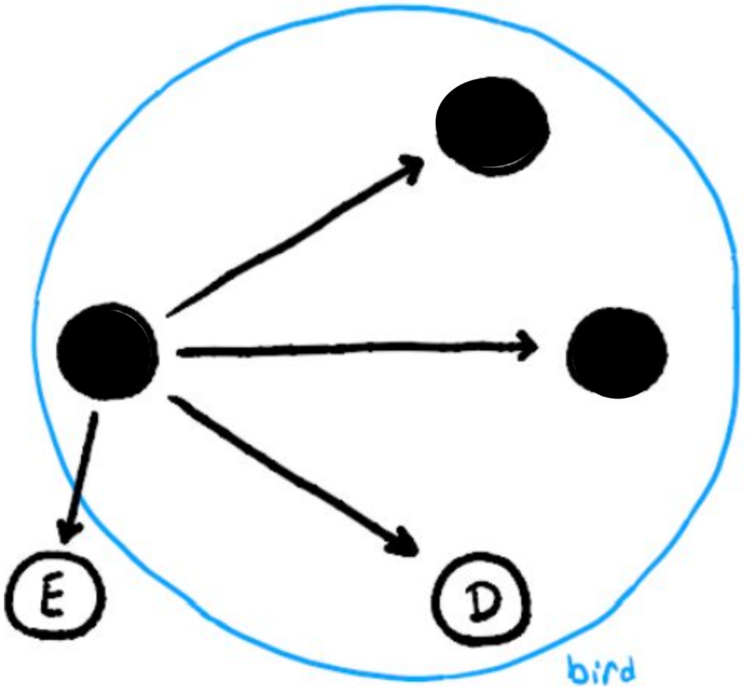
- Associative Learning
  - “Neurons that fire together wire together”
  - Backpropagation
- Simplify our model
  - One proposition
  - Binary classification
  - Only excitatory connections



Update via Associative Learning
(Negative example: Platypus)



Update via Associative Learning
(Positive Example: Duck)



Q&A

References:

Balkenius, C. and Gärdenfors, P., 1991, April. **Nonmonotonic Inferences in Neural Networks**. In *KR* (pp. 32-39).

Blutner, R. 2004. **Nonmonotonic inferences and neural networks**. In *Information, Interaction and Agency*. Springer. 203–234.

Hansson, S. O. 2017. **Logic of Belief Revision**. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2017 edition

Leitgeb, H. 2004. **Inference on the low level: An investigation into deduction, nonmonotonic reasoning, and the philosophy of cognition**, volume 30. *Springer Science & Business Media*.

Leitgeb, H. 2005. **From logic to neural networks and back**.

Leitgeb, H. 2018. **Neural network models of conditionals. In Introduction to Formal Philosophy**. Springer. 147–176.

McCulloch, W. S., and Pitts, W. 1943. **A logical calculus of the ideas immanent in nervous activity**. *The bulletin of mathematical biophysics* 5(4):115–133.

Siegelmann, H.T. and Sontag, E.D., 1995. **On the computational power of neural nets**. *Journal of computer and system sciences*, 50(1), pp.132-150.

Weiss, G., Goldberg, Y. and Yahav, E., 2018. **On the practical computational power of finite precision rnns for language recognition**. arXiv preprint arXiv:1805.04908.