Neuro-Symbolic Modeling: Introduction

THE UNIVERSITY OF UTAH

Does successful prediction imply understanding?

Date: May 28, 585 BCE

Time: Late afternoon

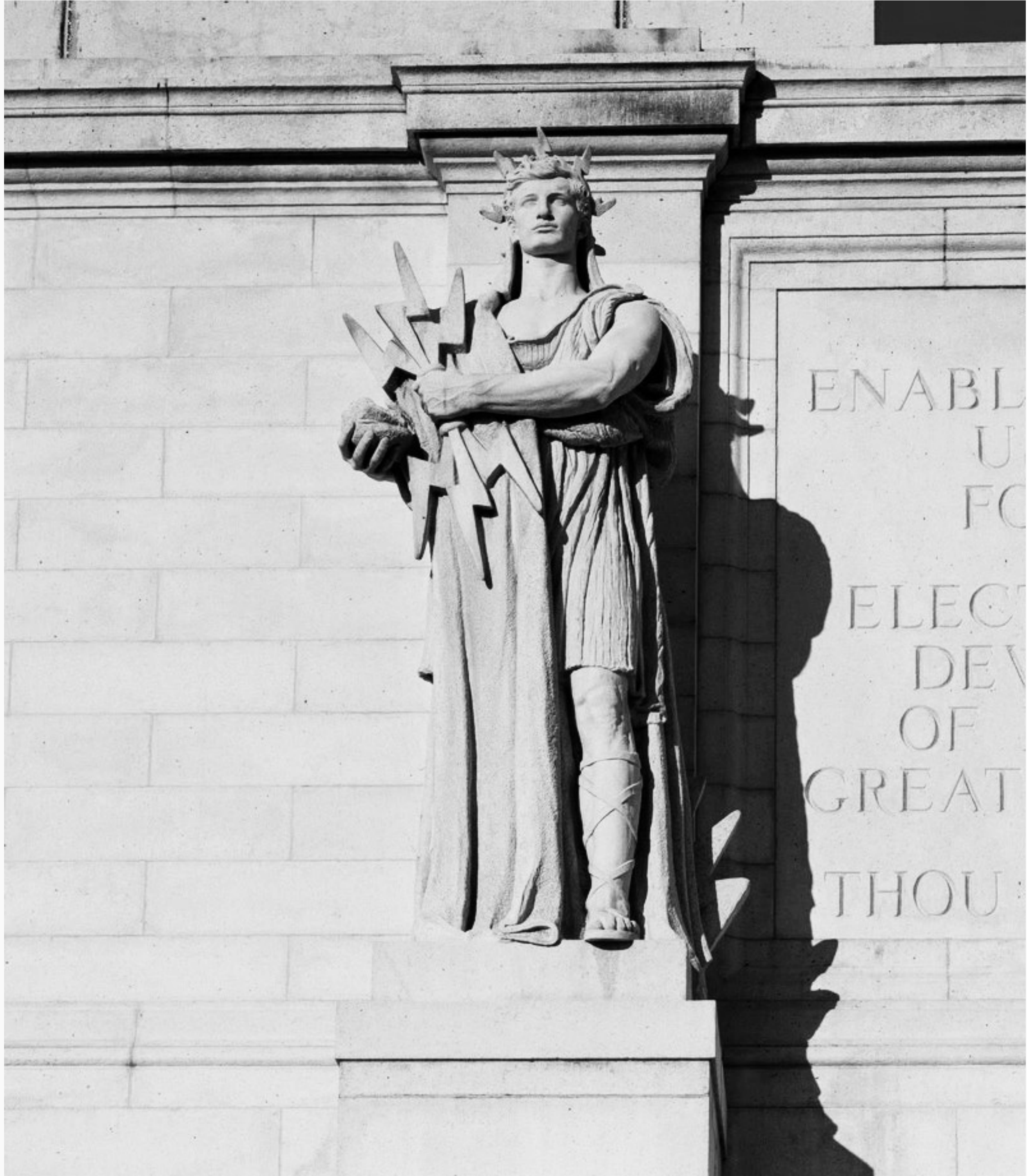
Location: Ancient Greece

Thales of Miletus

mathematician, astronomer, philosopher

One of the earliest known successful eclipse predictions

- Possibly learned how to do so from the Babylonian astronomers



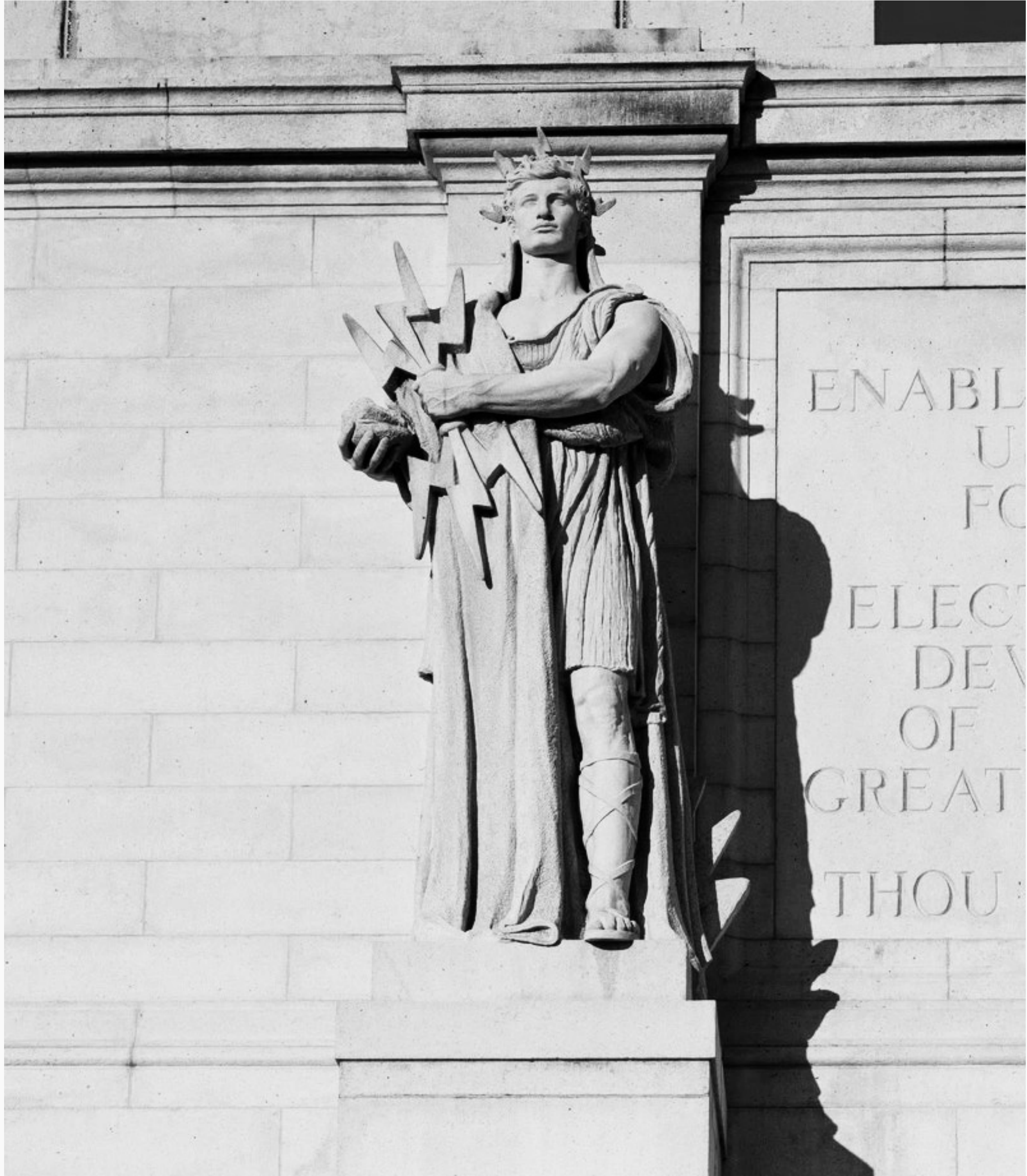
Thales of Miletus

mathematician, astronomer, philosopher

One of the earliest known successful eclipse predictions

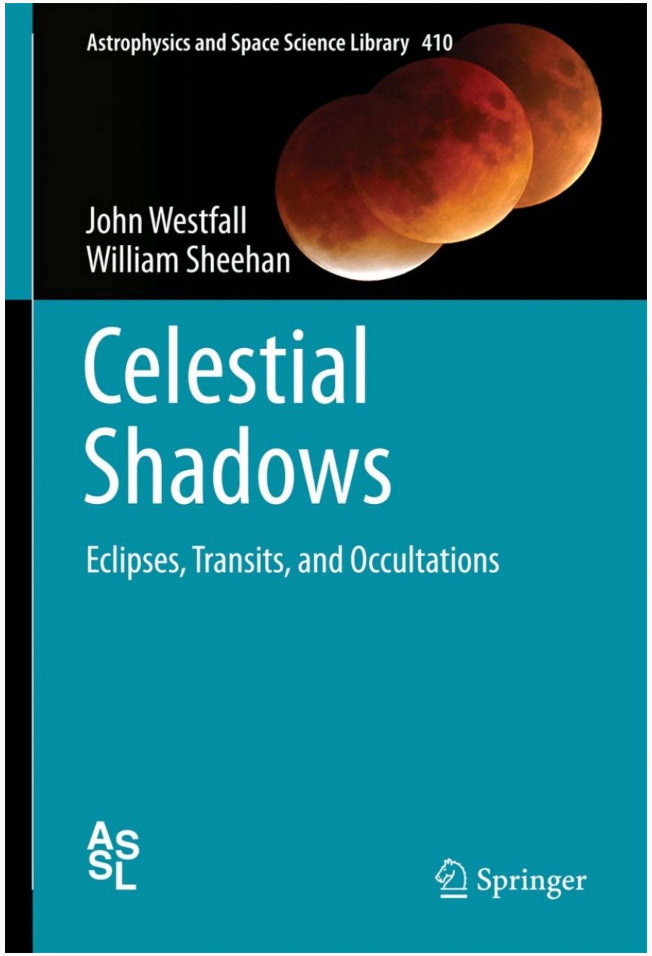
Possibly learned how to do so from the Babylonian astronomers

Did Thales (and Babylonians) understand how eclipses work?



"The recognition that solar eclipses are caused by the Moon coming between the Earth and the Sun did not actually come until over a century [after Thales...].

Thus Thales cannot have predicted an eclipse in any modern sense."



Westfall, John, and William Sheehan. *Celestial Shadows: Eclipses, Transits, and Occultations*. Vol. 410. Springer, 2014.

Prediction sans understanding

Thales & co “data mined” celestial observations to find cycles in lunar and solar eclipses…

but

…they had no clue about the positions of the earth, the moon and the sun!

Prediction sans understanding

Thales & co “data mined” celestial observations to find cycles in lunar and solar eclipses…	but	…they had no clue about the positions of the earth, the moon and the sun!
Finding patterns in data can lead to seemingly accurate predictions…	but	…accuracy in prediction does not signal understanding

Prediction sans understanding

Thales & co “data mined” celestial observations to find cycles in lunar and solar eclipses… but …they had no clue about the positions of the earth, the moon and the sun!

Finding patterns in data can lead to seemingly accurate predictions… but …accuracy in prediction does not signal understanding

Spurious patterns → A recipe for pathological failures

We have increasingly sophisticated pattern recognition machines today

Neural networks the default modeling tool today

They have demonstrated remarkable successes

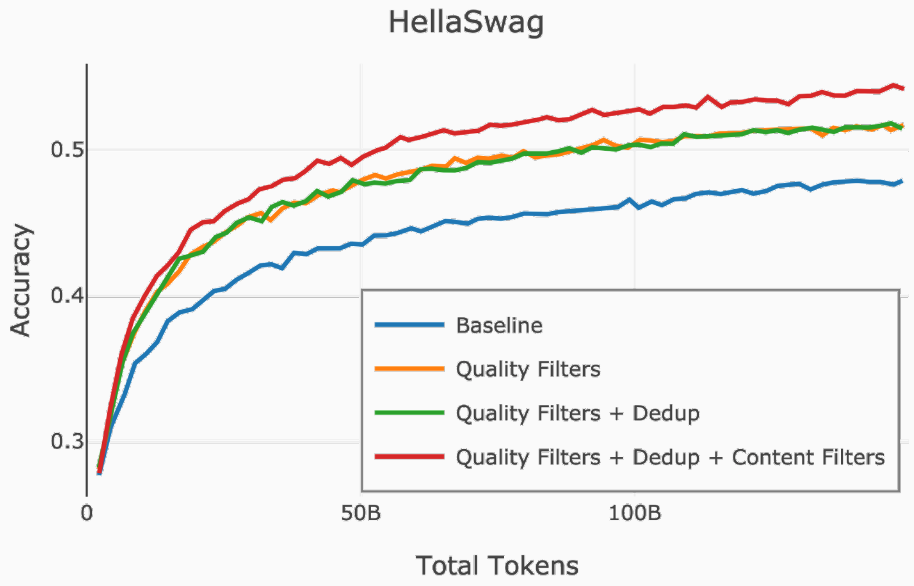
Neural networks the default modeling tool today

They have demonstrated remarkable successes

But…

- They need a lot of data!

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl	web pages	9,812	3,734	1,928	2,479
GitHub	code	1,043	210	260	411
Reddit	social media	339	377	72	89
Semantic Scholar	papers	268	38.8	50	70
Project Gutenberg	books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7	4.3
Total		**11,519**	**4,367**	**2,318**	**3,059**



Luca Soldaini et al. "Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research." ACL 2024

Neural networks the default modeling tool today

They have demonstrated remarkable successes

But…

- They need a lot of data!
- Some results are not easy to explain.

  Eight of the top ten models on the SUPERGLUE benchmark suite for text understanding outperform the human baseline! *What does that mean?*

SuperGLUE GLUE

Leaderboard Version: **2.0**

Rank	Name	Model	Score
1	Inspur Cloud	Hairuo	91.4
2	JDExplore d-team	Vega v2	91.3
3	Liam Fedus	ST-MoE-32B	91.2
4	Microsoft Alexander v-team	Turing NLR v5	90.9
5	ERNIE Team - Baidu	ERNIE 3.0	90.6
6	Yi Tay	PaLM 540B	90.4
7	Zirui Wang	T5 + UDG, Single Model (Google Brain)	90.4
8	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	90.3
9	SuperGLUE Human Baselines	SuperGLUE Human Baselines	89.8
10	T5 Team - Google	T5	89.3

Screenshot from August 26, 2024

Failures in reasoning and understanding

Visual question answering

From https://visualqa.org, uses Pythia v0. 1: the winning entry to the VQA challenge 2018 [Jiang, Y et al 2018].

Visual question answering

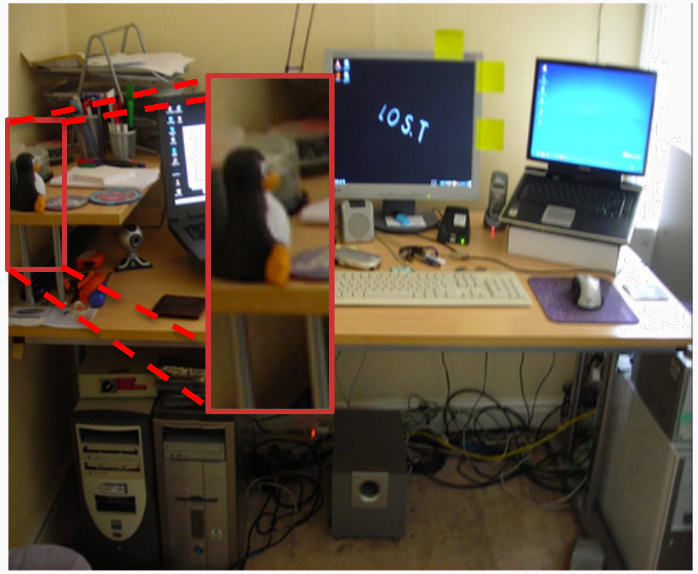
Where is the penguin?

Predicted top-5 answers with confidence:

on desk	37.025%
desk	15.443%
on table	12.358%
nowhere	6.715%
floor	4.579%

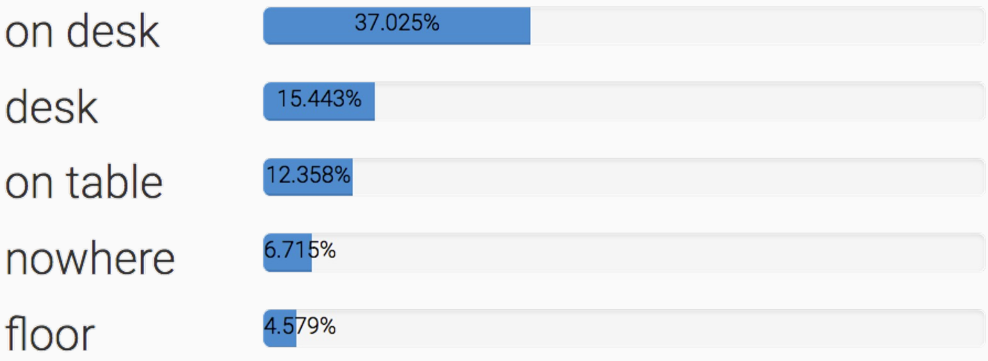
From https://visualqa.org, uses Pythia v0. 1: the winning entry to the VQA challenge 2018 [Jiang, Y et al 2018].

Visual question answering



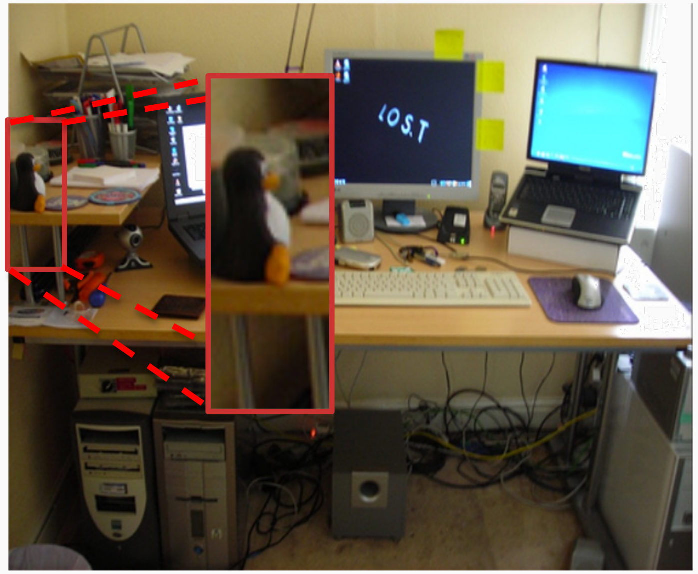
Where is the penguin?

Predicted top-5 answers with confidence:



From https://visualqa.org, uses Pythia v0. 1: the winning entry to the VQA challenge 2018 [Jiang, Y et al 2018].

Visual question answering



Where is the penguin?

Predicted top-5 answers with confidence:

on desk	37.025%
desk	15.443%
on table	12.358%
nowhere	6.715%
floor	4.579%

But can they really ‘read’ images?

Is there a penguin in the image?

Predicted top-5 answers with confidence:

no	100.000%
yes	0.000%

From https://visualqa.org, uses Pythia v0. 1: the winning entry to the VQA challenge 2018 [Jiang, Y et al 2018].

Can neural networks ‘read’ images?

Where is the penguin?

Predicted top-5 answers with confidence:

on desk	37.025%
desk	15.443%
on table	12.358%
nowhere	6.715%

```
Only one of these answers can be correct.
```

But the model is devoid of any such “theory”

Predicted top-5 answers with confidence:

no	100.000%
yes	0.000%

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is **1** letter circled in the image.

Example uses Google Gemini, Aug 26, 2024

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is **1** letter circled in the image.

Almost impressive but not quite right. The letter 'o' isn't circled

Example uses Google Gemini, Aug 26, 2024

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is **1** letter circled in the image.

Is the letter 'c' circled?

Yes, the letter 'c' is circled in the image.

Example uses Google Gemini, Aug 26, 2024

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is **1** letter circled in the image.

Given this claim…

Is the letter 'c' circled?

Yes, the letter 'c' is circled in the image.

Example uses Google Gemini, Aug 26, 2024

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is 1 letter circled in the image.

Given this claim, at most one of these can be true

Is the letter 'c' circled?

Yes, the letter 'c' is circled in the image.

Example uses Google Gemini, Aug 26, 2024

But how about more recent models?

How many letters are circled?

The image shows the word "zoological" with the letter "o" circled. Therefore, there is 1 letter circled in the image.

Given this claim, at most one of these can be true

Is the letter 'c' circled?

But the model is devoid of any such "theory"

Yes, the letter 'c' is circled in the image.

Example uses Google Gemini, Aug 26, 2024

Natural language inference

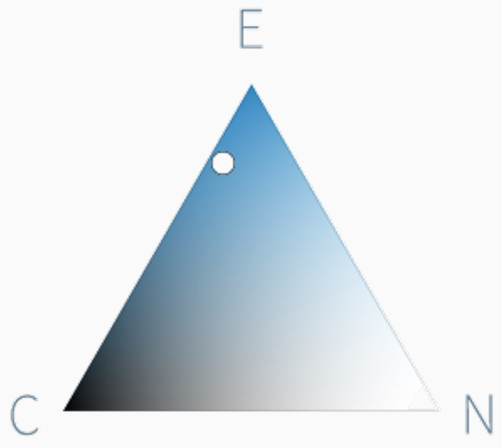
Premise Before it moved to Chicago, aerospace manufacturer Boeing was the largest company in Seattle.

Hypothesis Boeing is a Chicago-based aerospace manufacturer.

Natural language inference

Premise Before it moved to Chicago, aerospace manufacturer Boeing was the largest company in Seattle.

Hypothesis Boeing is a Chicago-based aerospace manufacturer.



Judgment	Probability
Entailment	75.6%
Contradiction	19.9%
Neutral	4.5%

It is quite likely that the premise entails the hypothesis.

https://demo.allennlp.org/textual-entailment/

Can neural networks understand text?

P John is on a train to Berlin.

H John is traveling to Berlin.

Z John is having lunch in Berlin.

Can neural networks understand text?

P	John is on a train to Berlin.
H	John is traveling to Berlin.
Z	John is having lunch in Berlin.

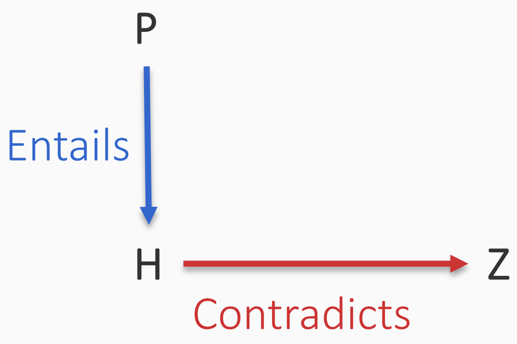
$$P \xrightarrow{\text{Entails}} H$$

Can neural networks understand text?

P John is on a train to Berlin.

H John is traveling to Berlin.

Z John is having lunch in Berlin.

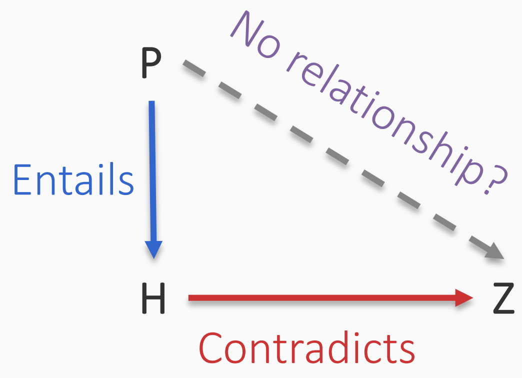


Can neural networks understand text?

P John is on a train to Berlin.

H John is traveling to Berlin.

Z John is having lunch in Berlin.

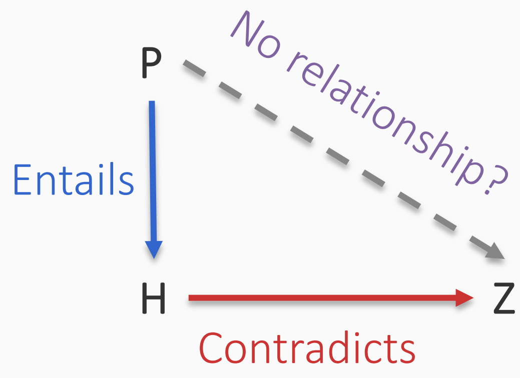


Can neural networks understand text?

P John is on a train to Berlin.

H John is traveling to Berlin.

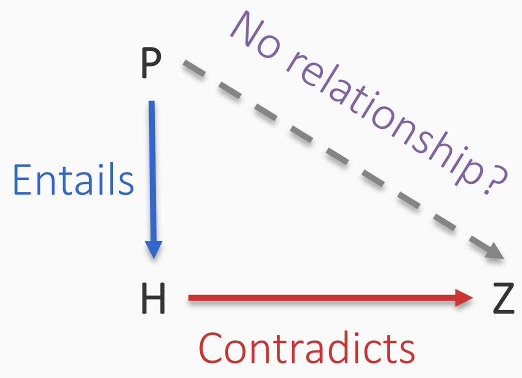
Z John is having lunch in Berlin.



The same system cannot simultaneously hold these three beliefs!

Can neural networks understand text?

P	John is on a train to Berlin.
H	John is traveling to Berlin.
Z	John is having lunch in Berlin.



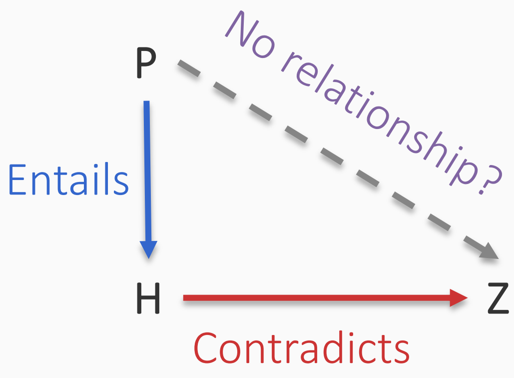
The same system cannot simultaneously hold these three beliefs!

Violates this invariant

```
If P entails H and H contradicts Z,
    then P contradicts Z
```

Can neural networks understand text?

P	John is on a train to Berlin.
H	John is traveling to Berlin.
Z	John is having lunch in Berlin.



The same system cannot simultaneously hold these three beliefs!

Violates this invariant

```
If P entails H and H contradicts Z,
    then P contradicts Z
```

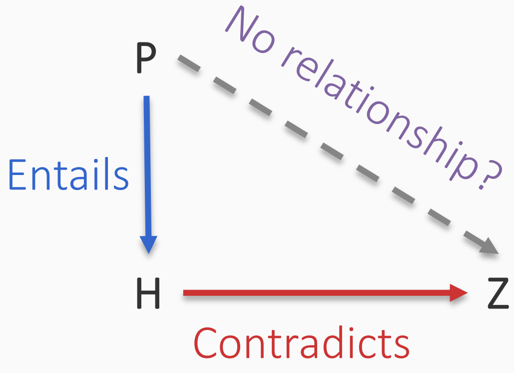
A BERT-based model that gets ~90% on benchmark data violates this invariant on 46% of a large collection of sentence triples.

Can neural networks understand text?

P John is on a train to Berlin.

H John is traveling to Berlin.

Z John is having lunch in Berlin.



If *P entails H* and *H contradicts Z*,
then P contradicts Z

Can today's neural networks use such "theory" in the form of invariant knowledge?

Are we modeling problems in their full richness?

Or: are we modeling for benchmark test sets?

Challenge: Modeling strategies that depend on or expose a theory to reduce data dependence

(or at least assume the existence of a theory)

– Otherwise, we are just guessing equations involving high dimensional spaces, without understanding what they mean

– Thales, versus a modern understanding of eclipses

Precise predictions via curve fitting versus a philosophy of the underlying phenomenon

The problem with purely data-driven machine learning

Large models trained on large amounts of data can functionally approximate the intelligent behavior that led to the creation of the data

- Eg: Language models can produce token sequences that resemble language in style and content

But explanation and understanding? We need systems that…

- …can be controlled via mechanisms other than just “add more data”
- …can provide insight into their reasoning processes

A trip to the past

Good Old Fashioned Artificial Intelligence (GOFAI)

Symbolic Artifical Intelligence

An agenda for artificial intelligence that focuses on knowledge representation and reasoning

Uses symbols (i.e. *discrete* labels), rules, and logic to represent knowledge

- Typically hand crafted rules

Well understood algorithms can reason about it

Rich human-auditable representations of the semantics and behavior of the programs

Historically a dominant way to think about AI

1950s: The Dartmouth Summer Research Project, John McCarthy's work (e.g. Advice Taker)

1960s-70s: The development of rule-based systems

Two examples:

- ELIZA (Weizenbaum 1966): A chatbot that simulated a psychotherapist
- PARRY (Colby 1972): A chatbot that simulated a paranoid schizophrenic

1970s-1980s: The rise of logic programming (e.g., Prolog)

A representative of this approach: Reasoning programs

Reasoning programs have various kinds of inputs and outputs (arrays to represent images, sequences for utterances, parse trees for sentences, etc)

McCarthy, John, and Patrick Hayes. "Some philosophical problems from the standpoint of artificial intelligence." (1969).

A representative of this approach: Reasoning programs

Reasoning programs have various kinds of inputs and outputs (arrays to represent images, sequences for utterances, parse trees for sentences, etc)

All input/output situations and internal program states are represented by *symbolic expressions in a formal logic*

McCarthy, John, and Patrick Hayes. "Some philosophical problems from the standpoint of artificial intelligence." (1969).

A representative of this approach: Reasoning programs

Reasoning programs have various kinds of inputs and outputs (arrays to represent images, sequences for utterances, parse trees for sentences, etc)

All input/output situations and internal program states are represented by *symbolic expressions in a formal logic*

- Data structures, agent goals and sub-goals, rules governing behavior

McCarthy, John, and Patrick Hayes. "Some philosophical problems from the standpoint of artificial intelligence." (1969).

A representative of this approach: Reasoning programs

Reasoning programs have various kinds of inputs and outputs (arrays to represent images, sequences for utterances, parse trees for sentences, etc)

All input/output situations and internal program states are represented by ***symbolic expressions in a formal logic***

- Data structures, agent goals and sub-goals, rules governing behavior

The program is a *deduction engine* that

- tries to find strategies of action that it can prove will solve a problem, and
- on finding one, executes it

McCarthy, John, and Patrick Hayes. "Some philosophical problems from the standpoint of artificial intelligence." (1969).

An example from McCarthy (1959)

The situation: "Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?

Answer: "walk to the car and drive the car to the airport"

How is this represented symbolically?

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

"Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?
Answer: "walk to the car and drive the car to the airport"

First, the facts of the problem

$at(I, desk)$ $want(at(I, airport))$

$at(desk, home)$

$at(car, home)$

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

"Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?
Answer: "walk to the car and drive the car to the airport"

First, the facts of the problem

$at(I, desk)$ $want(at(I, airport))$

$at(desk, home)$

$at(car, home)$

$at(home, county)$

$at(airport, county)$

Where did these come from?

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

"Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?
Answer: "walk to the car and drive the car to the airport"

First, the facts of the problem

$at(I, desk)$ $want(at(I, airport))$

$at(desk, home)$

$at(car, home)$

$at(home, county)$

$at(airport, county)$

The "at" predicate is transitive

$at(x, y), at(y, z) \rightarrow at(x, z)$

This holds for any x, y, z.
But where did this rule come from?

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

"Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?
Answer: "walk to the car and drive the car to the airport"

First, the facts of the problem

$$at(I, desk) \qquad want(at(I, airport))$$

$$at(desk, home)$$

$$at(car, home)$$

$$at(home, county)$$

$$at(airport, county)$$

The "at" predicate is transitive

$$at(x, y), at(y, z) \rightarrow at(x, z)$$

Define feasibility of walking and driving

$$walkable(x), at(y, x), at(z, x), at(I, y) \rightarrow can(go(y, z, walking))$$

$$drivable(x), at(y, x), at(z, x), at(car, y), at(I, car) \rightarrow can(go(y, z, driving))$$

Something to consider: Do we really think about this so explicitly?

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

"Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also." What should I do?
Answer: "walk to the car and drive the car to the airport"

First, the facts of the problem

$at(I, desk)$ $want(at(I, airport))$

$at(desk, home)$

$at(car, home)$

$at(home, county)$

$at(airport, county)$

$$\boxed{\begin{array}{c} walkable(home) \\ drivable(county) \end{array}}$$

This is not stated in the question, but we have to assume this. Which means, it needs to be explicitly represented. Both a strength and a weakness.

The "at" predicate is transitive

$$at(x, y), at(y, z) \rightarrow at(x, z)$$

Define feasibility of walking and driving

$$walkable(x), at(y, x), at(z, x), at(I, y) \rightarrow can(go(y, z, walking))$$

$$drivable(x), at(y, x), at(z, x), at(car, y), at(I, car) \rightarrow can(go(y, z, driving))$$

McCarthy, John, and Patrick Hayes. "Programs with Common Sense." (1953).

An example from McCarthy (1959)

“Assume that I am seated at my desk at home and I wish to go to the airport. My car is at my home also.” What should I do? Answer: “walk to the car and drive the car to the airport”

First, the facts of the problem

$$at(I, desk) \quad want(at(I, airport))$$

$$at(desk, home) \quad walkable(home)$$

$$at(car, home)$$

$$at(home, county) \quad drivable(county)$$

$$at(airport, county)$$

The “at” predicate is transitive

$$at(x, y), at(y, z) \rightarrow at(x, z)$$

Define feasibility of walking and driving

$$walkable(x), at(y, x), at(z, x), at(I, y) \rightarrow can(go(y, z, walking))$$

$$drivable(x), at(y, x), at(z, x), at(car, y), at(I, car) \rightarrow can(go(y, z, driving))$$

And this goes on for a couple of pages…

McCarthy, John, and Patrick Hayes. ”Programs with Common Sense." (1953).

The problem with GOFAI

Worked well, but *only* for a very small set of examples

Did not handle uncertainty and noise well

Took several years of human effort to create

And still did not generalize

Because too many hand-crafted rules

Hand-crafted rules do not really reflect how complex phenomena like language and vision work in practice!

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

Symbolic AI

Historically well developed and has deep algorithmic understanding

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design
- ✗ Do not easily work with easily stated rules about a problem

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy
- ✓ Naturally defined to use declaratively stated rules

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design
- ✗ Do not easily work with easily stated rules about a problem
- ✗ Does not really perform reasoning (and planning, search, or any recursive algorithmic behavior)

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy
- ✓ Naturally defined to use declaratively stated rules
- ✓ Well suited for reasoning (and planning, search, or any recursive algorithmic behavior)

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design
- ✗ Do not easily work with easily stated rules about a problem
- ✗ Does not really perform reasoning (and planning, search, or any recursive algorithmic behavior)
- ✗ Poor auditability of decision process

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy
- ✓ Naturally defined to use declaratively stated rules
- ✓ Well suited for reasoning (and planning, search, or any recursive algorithmic behavior)
- ✓ Transparent decision process

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design
- ✗ Do not easily work with easily stated rules about a problem
- ✗ Does not really perform reasoning (and planning, search, or any recursive algorithmic behavior)
- ✗ Poor auditability of decision process

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy
- ✓ Naturally defined to use declaratively stated rules
- ✓ Well suited for reasoning (and planning, search, or any recursive algorithmic behavior)
- ✓ Transparent decision process

What do we want: Best of both worlds

Neural networks versus Symbolic AI

Neural networks

The best pattern recognition engines today

- ✓ Handles uncertainty
- ✓ Use raw data to improve behavior
- ✓ Easy to design
- ✗ Do not easily work with easily stated rules about a problem
- ✗ Does not really perform reasoning (and planning, search, or any recursive algorithmic behavior)
- ✗ Poor auditability of decision process

Symbolic AI

Historically well developed and has deep algorithmic understanding

- ✗ Does not handle uncertainty
- ✗ Does not use raw data to improve behavior
- ✗ Difficult to design and deploy
- ✓ Naturally defined to use declaratively stated rules
- ✓ Well suited for reasoning (and planning, search, or any recursive algorithmic behavior)
- ✓ Transparent decision process

What do we want: Neuro-Symbolic Models

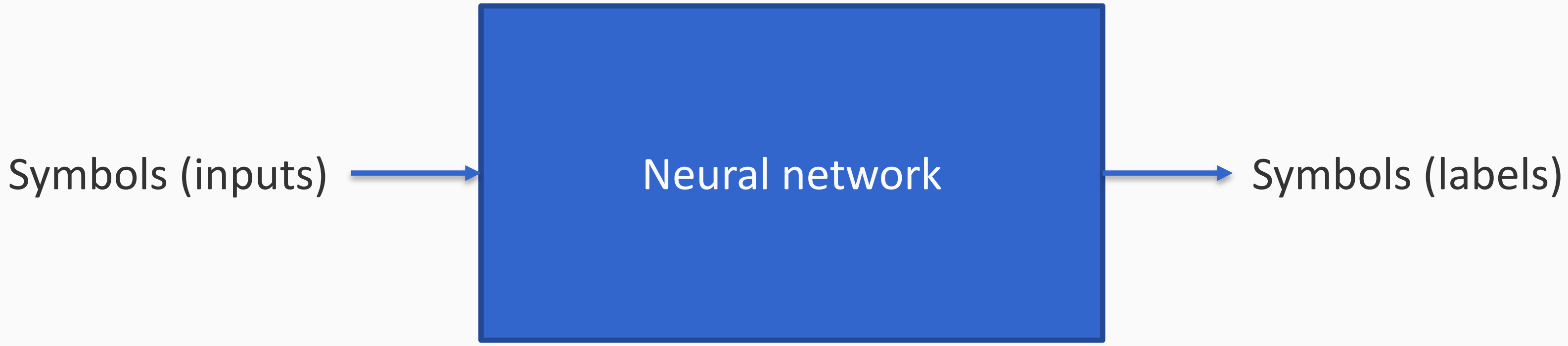
Neural network + symbolic programs

Neural networks involve vectors and differentiable functions

Rules are symbolic objects (i.e. consisting of discrete decisions)

How do they interface with each other?

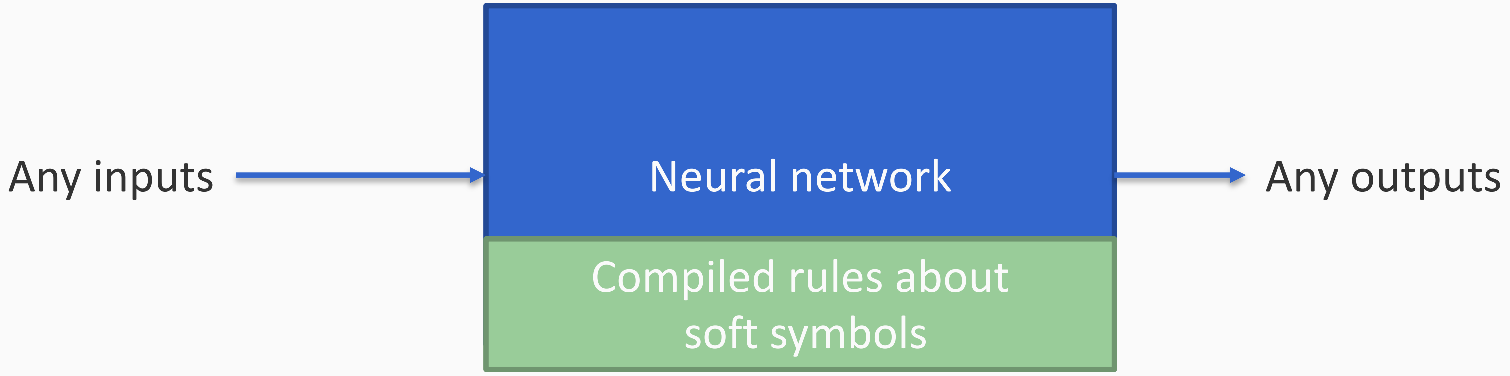
Neuro-symbolic interfaces



Convert symbolic inputs into vectors, operate with them and produce symbolic outputs

Standard neural networks do this

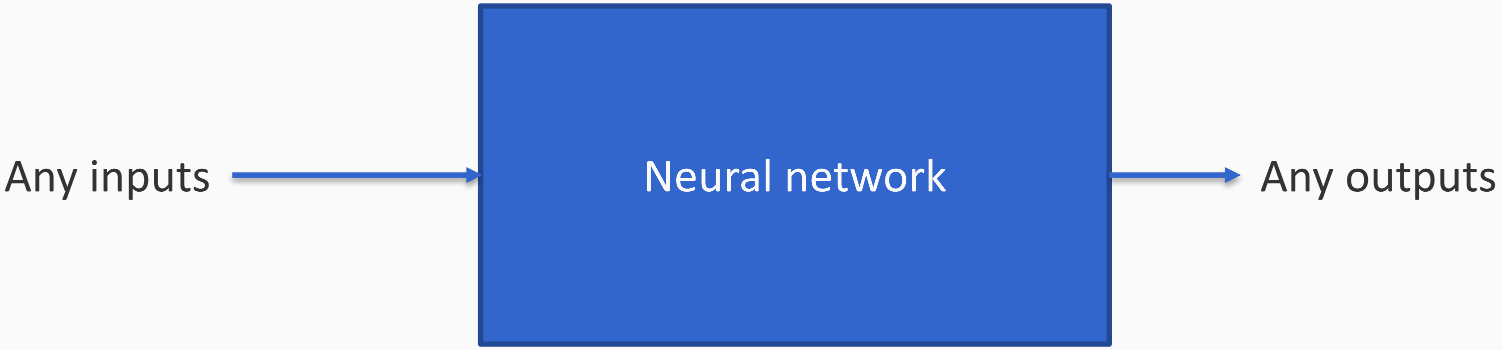
Neuro-symbolic interfaces



Treat some nodes within a neural network as soft symbols

Compile symbolic rules (i.e. constraints) into neural network submodules about these symbols to augment the network architecture

Neuro-symbolic interfaces

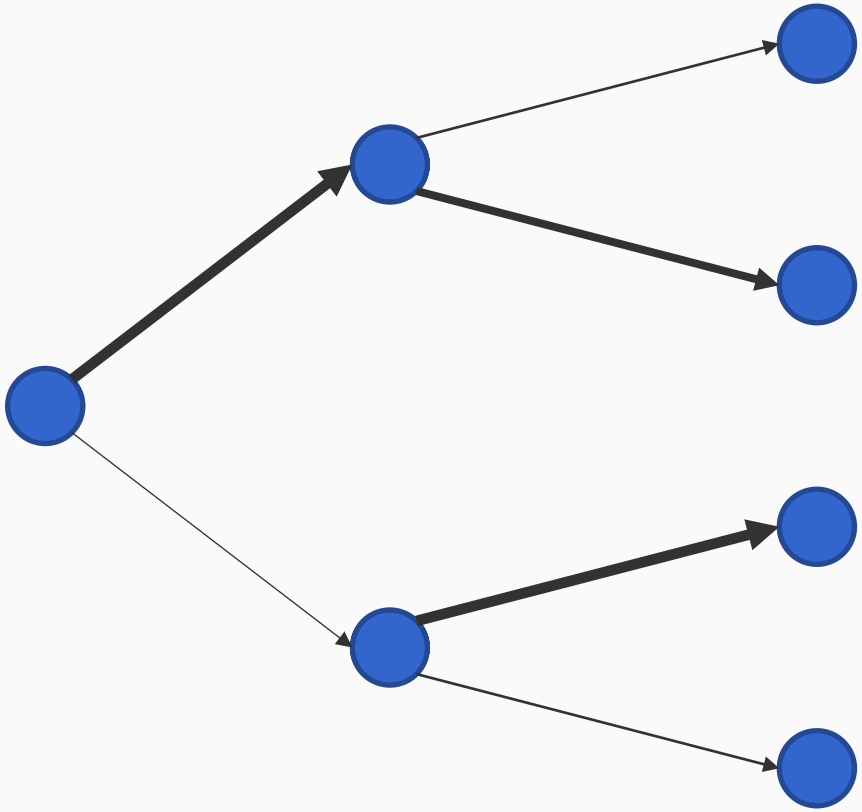


Training: data loss + compiled(rules)

Symbolic rules (about outputs or both inputs and outputs) compiled into a form that augments the training process

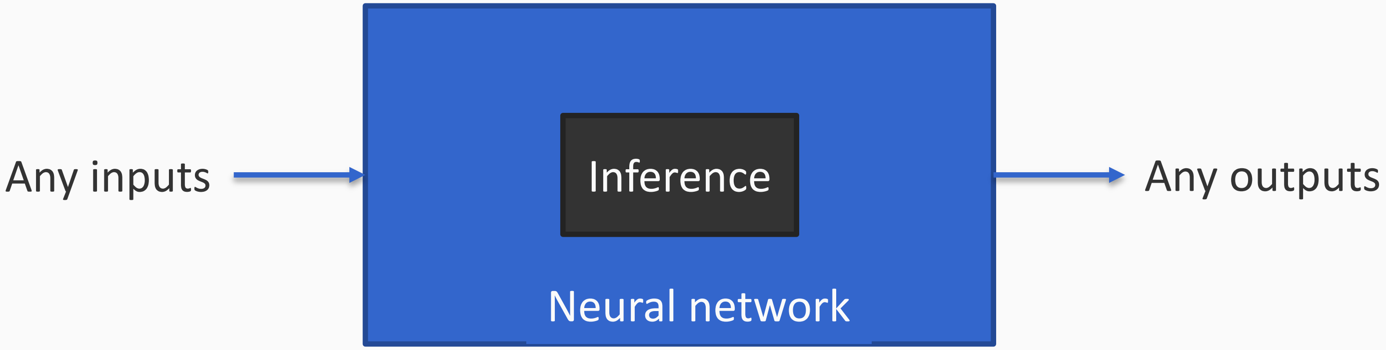
Implicit claim: Neural networks can internally represent the rules. *Is this valid?*

Neuro-symbolic interfaces



Use a neural network to guide a program that navigates a symbolic space by scoring paths (e.g. game playing, combinatorial inference, etc)

Neuro-symbolic interfaces



Symbolic reasoning inside a neural network

Differentiability is an issue.

Neuro-symbolic interfaces

Convert symbolic inputs into vectors, operate with distributions over labels

Apply probabilistic inference (with symbolic knowledge) over the top of network outputs to produce final outputs: *Structured prediction*

What we will cover this semester

Semester overview

Part 0: Introduction

- Review of neural networks
  - The computation graph abstraction and gradient-based learning
  - Special neural networks (e.g. transformers)
- Review of symbolic logic
  - propositional logic and SAT
  - tractable representations
  - knowledge compilation

Semester overview

Part 1: The "logic-as-loss" approach

- The idea that logic can be compiled into loss functions
- Two high level strategies for "logic-as-loss"
  - Weighted model counting, semantic loss
  - Soft logic, t-norms
- Strengths and limitations of this strategy

Semester overview

Part 2: The “logic-as-network” approach

- The idea that logic can be compiled into neural networks
- Using soft logic for this
- Strengths and limitations of this strategy

Semester overview

Part 2: The “structured prediction” approach

- Structured prediction and training
- MAXSAT based methods
- Integer programming based methods
- Differentiable combinatorial optimization

Semester overview

Part 3: The “Reinforcement Learning” approach

- Reinforcement learning introduction
- The REINFORCE algorithm
- Applications with black-box symbolic programs

Semester overview

Part 4: Case studies with neuro-symbolic modeling

– We will encounter several examples during the previous sections

  +

– Incorporating human preference feedback in large language models

  And derive standard named algorithms as special cases)

– Your favorite topic

  Let me know