# Symbolic AI

Andre Freitas

Photo by Vasilyev Alexandr

# Acknowledgements

Based on the slides of Richard Evans, Edward Grefenstette

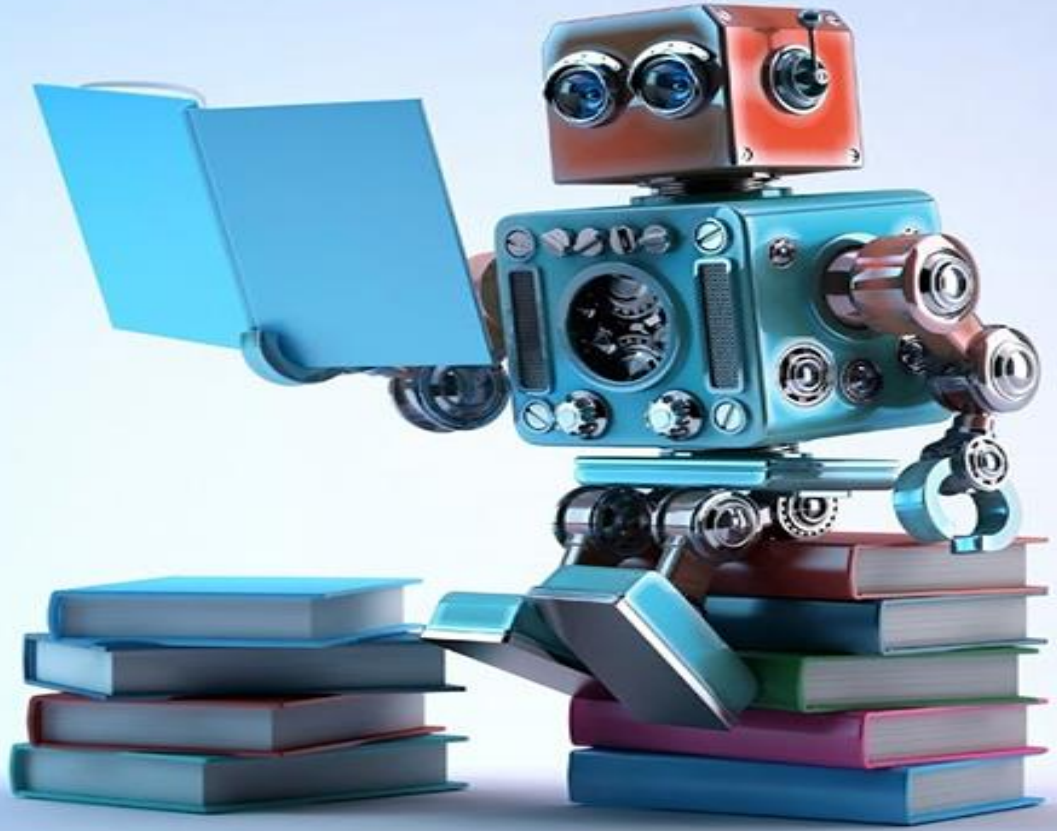
Evans, Richard, and Edward Grefenstette. "Learning explanatory rules from noisy data." *Journal of Artificial Intelligence Research*61 (2018): 1-64.

Thanks to Marco Valentino for some of the support slides.

MANCHESTER 1824
The University of Manchester

# Today

- We will introduce how Symbolic Models interact with Machine Learning Models.

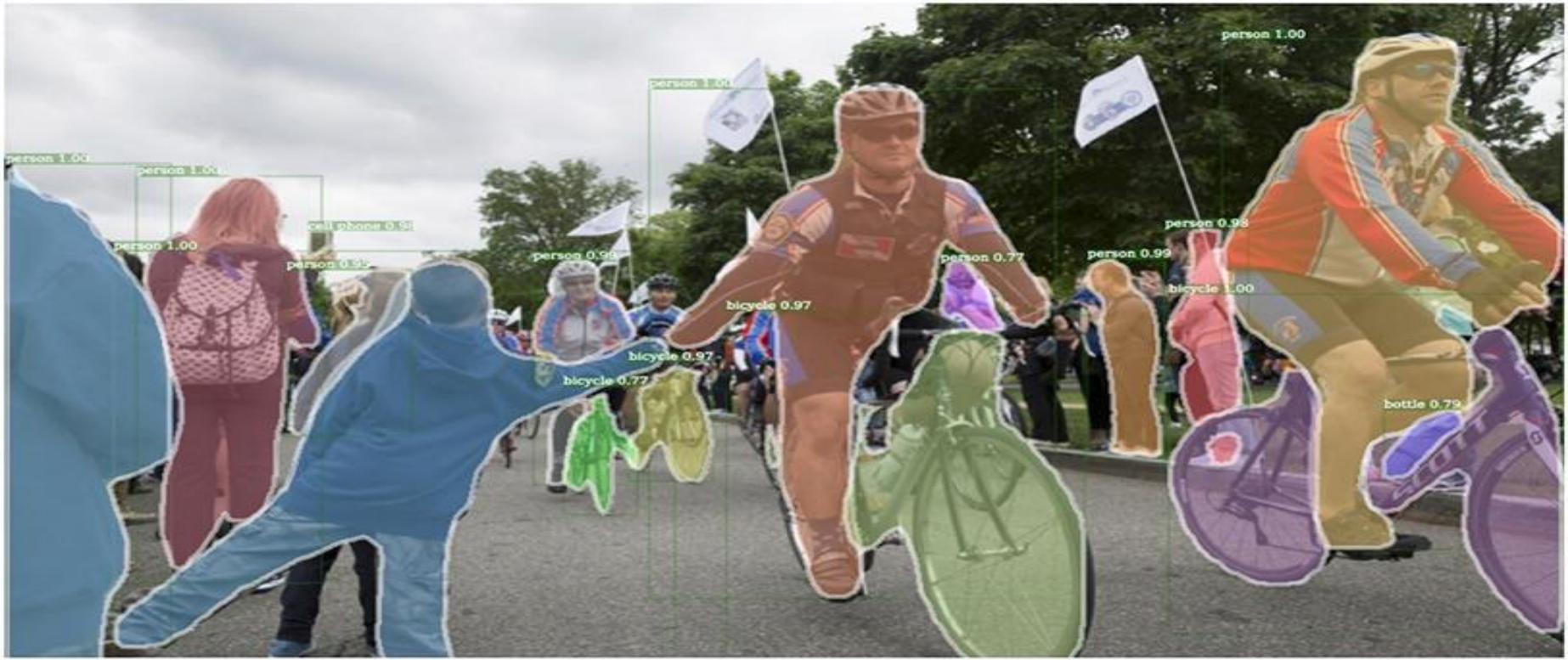


# Supervised Learning

MANCHESTER 1824
The University of Manchester

$Y($ I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet. $)=C$

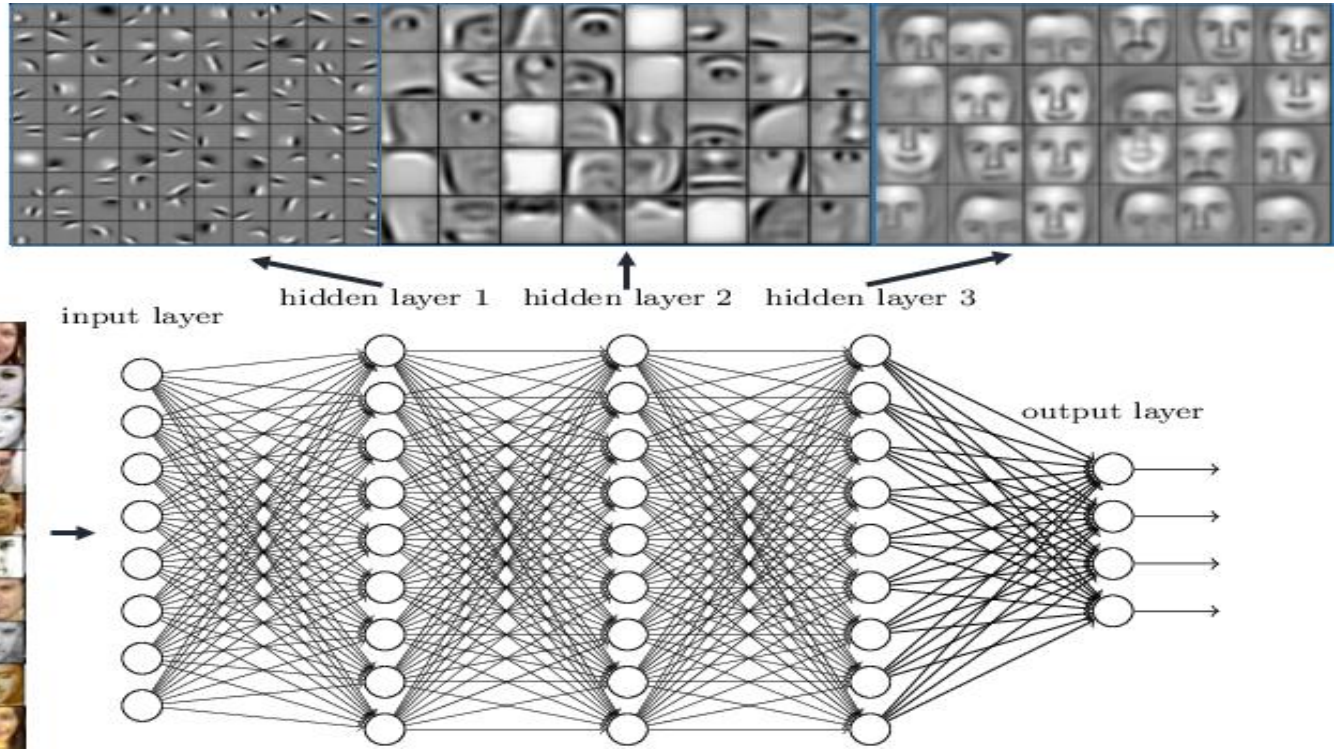
MANCHESTER
1824
The University of Manchester



MANCHESTER 1824
The University of Manchester

# Deep Neural Networks

Deep neural networks learn hierarchical feature representations



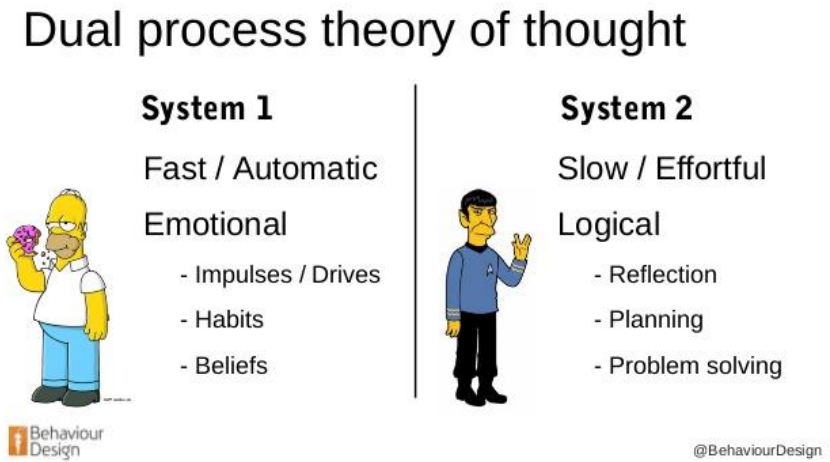
MANCHESTER 1824
The University of Manchester

https://www.youtube.com/watch?v=Ilg3gGewQ5U

# Dual Process Theory

Suppose you are playing football. The ball arrives at your feet, and you decide to pass it to the unmarked striker. What seems like one simple action requires **two different kinds of thought**.

- **Intuitive perceptual thinking:** you **recognise** that there is a football at your feet. You cannot easily articulate how you come to know that there is a ball at your feet, you **just see** that it is there.
- **Conceptual thinking:** you **decide** to pass the ball to a particular striker. Your decision is tied to a **justification** - the reason you passed the ball to the striker is because she was unmarked

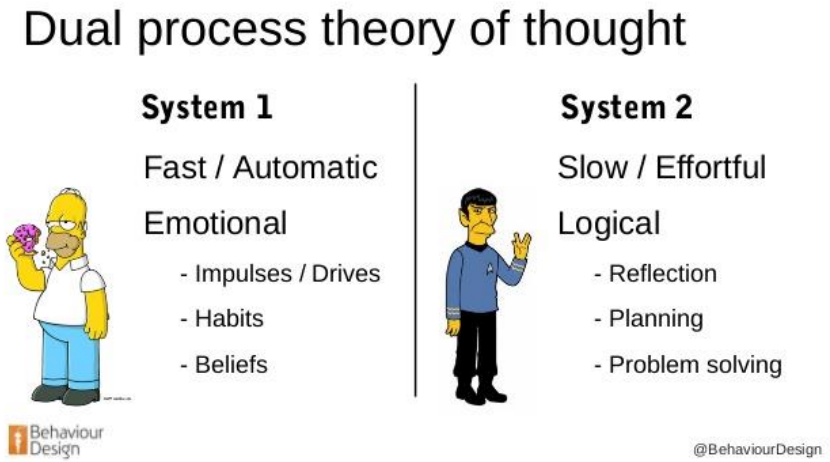



https://deepmind.com/blog/learning-explanatory-rules-noisy-data/

# Dual Process Theory

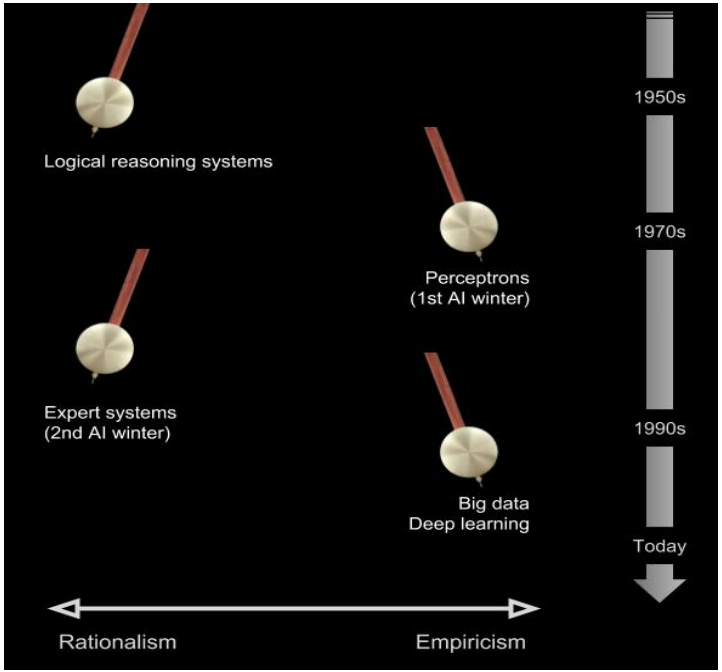
This distinction corresponds to **two different approaches** to machine learning:

- **Deep Learning** (System 1): concentrates on intuitive perceptual thinking
- **Symbolic Program Synthesis** (System 2): focuses on conceptual, rule-based thinking




https://deepmind.com/blog/learning-explanatory-rules-noisy-data/

# Rationalism vs Empiricism in AI

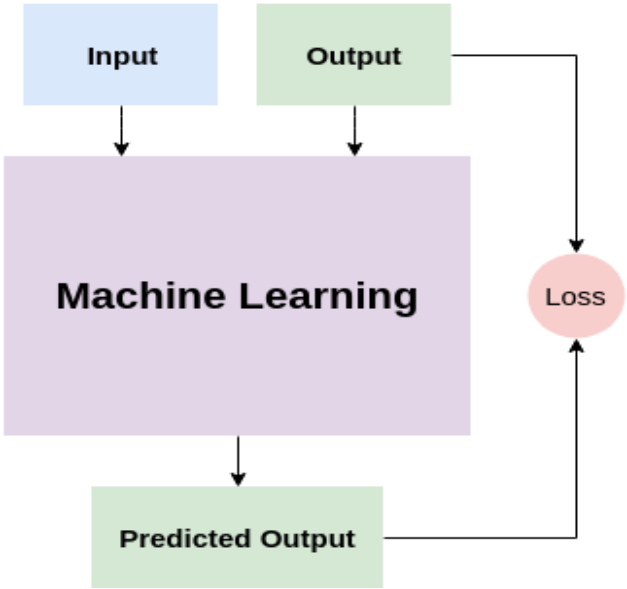



Rationalism is necessary yet insufficient

Empiricism is necessary yet insufficient

# Current limits of Deep Learning

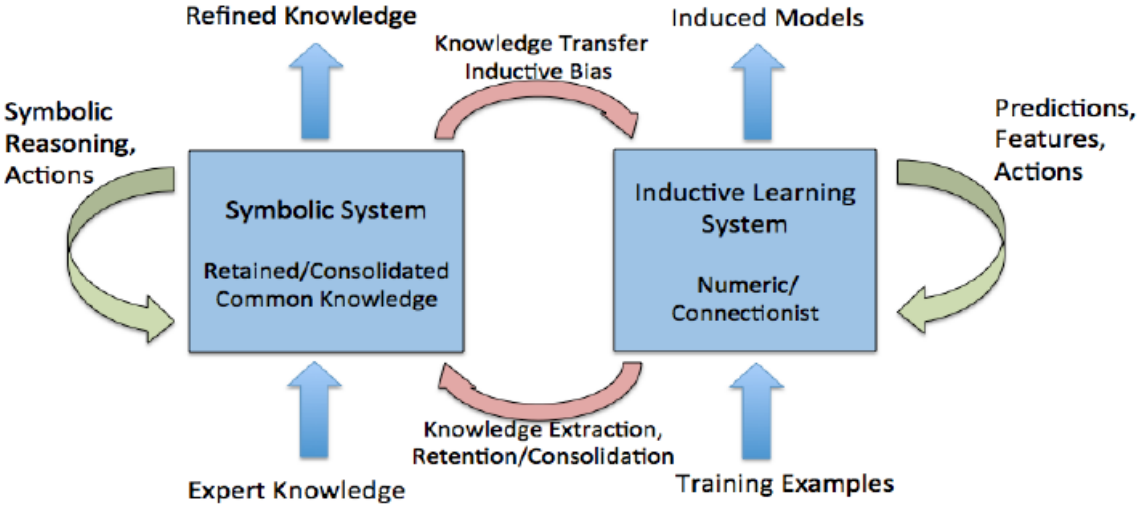
1. Deep learning thus far is data hungry
2. Deep learning thus far is shallow and has limited capacity for transfer
3. Deep learning thus far has no natural way to deal with hierarchical structure
4. Deep learning thus far is not sufficiently transparent
5. Deep learning thus far has not been well integrated with prior knowledge
6. Deep learning thus far cannot inherently distinguish causation from correlation
7. Deep learning presumes a largely stable world, in ways that may be problematic
8. Deep learning thus far works well as an approximation, but its answers often cannot be fully trusted
9. Deep learning thus far is difficult to engineer with




Marcus, Gary. "Deep learning: A critical appraisal." *arXiv preprint arXiv:1801.00631* (2018).

MANCHESTER 1824
The University of Manchester

# Intuitive vs Symbolic AI Systems




| | Intuitive | Symbolic |
|---|---|---|
| Explainability | Hard | Easy |
| Generalizing algebraic operations | Hard | Easy |
| Robustness to noise | Easy | Hard |
| Robustness to ambiguity | Easy | Hard |
| Robustness to mislabeling | Easy | Hard |

MANCHESTER 1824
The University of Manchester

# ∂ILP

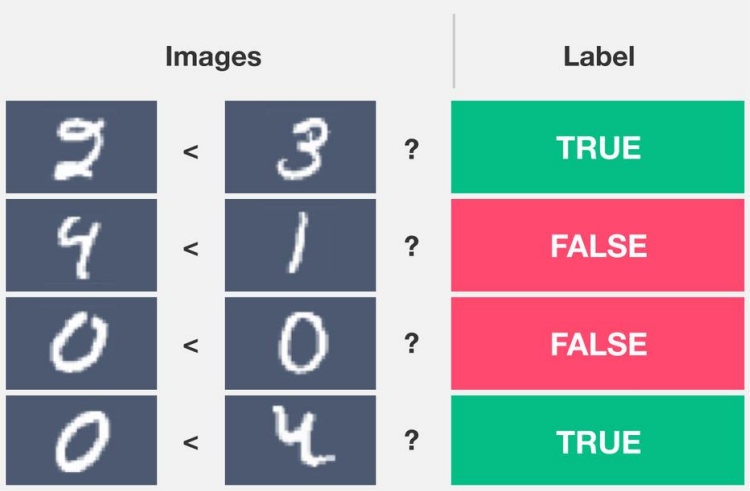
It is possible for systems to combine intuitive perceptual with conceptual interpretable reasoning!

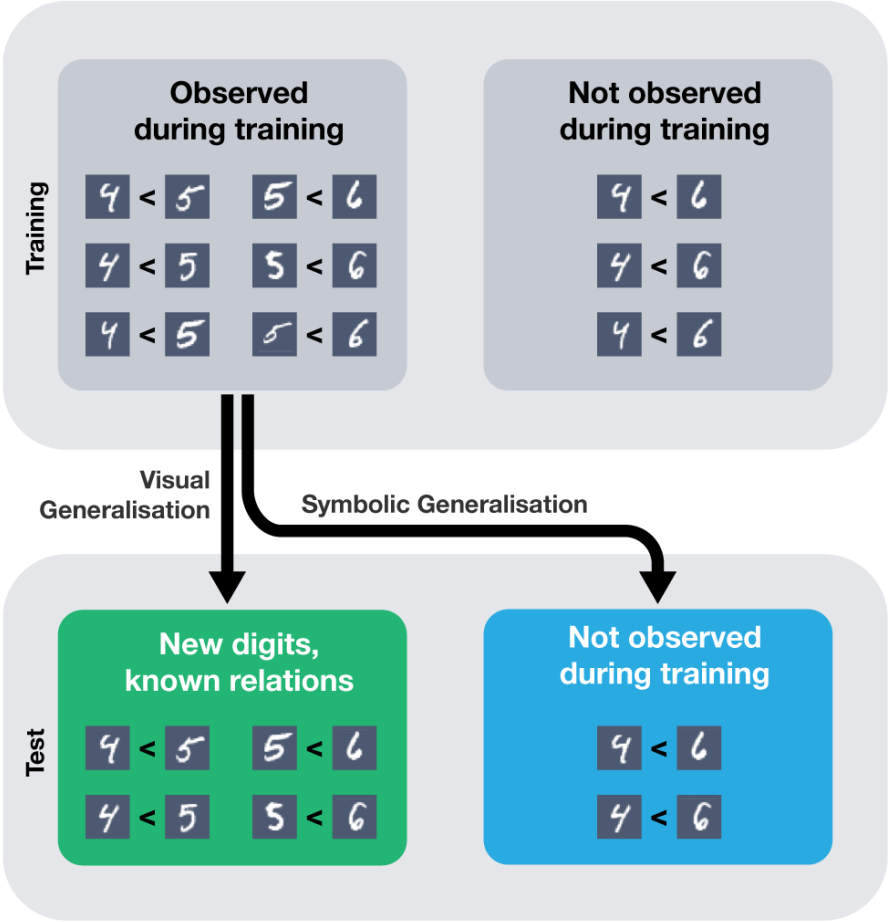
| | Deep Learning | Symbolic Program Synthesis | ∂ILP |
|---|---|---|---|
| Robust to noise | YES | NO | YES |
| Can learn from non-symbolic data | YES | NO | YES |
| Data efficient | NO | YES | YES |
| Interpretable | NO | YES | YES |

https://deepmind.com/blog/learning-explanatory-rules-noisy-data/

MANCHESTER 1824
The University of Manchester

∂ILP



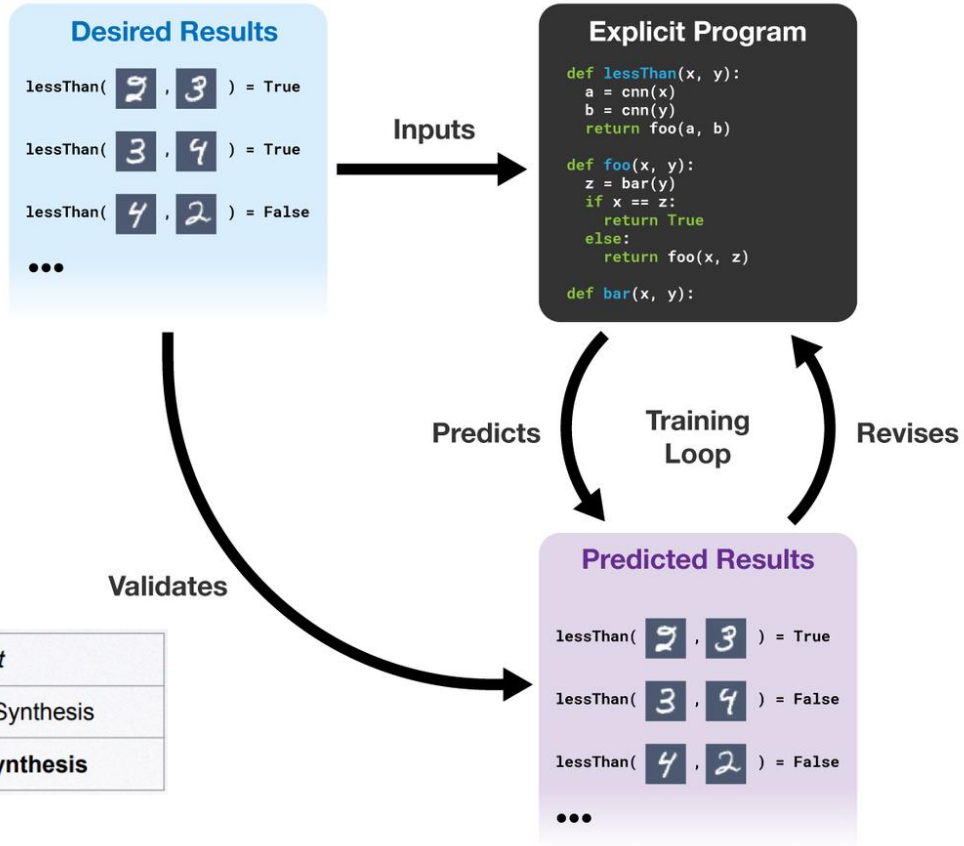




https://deepmind.com/blog/learning-explanatory-rules-noisy-data/

MANCHESTER 1824
The University of Manchester

# General Approach

- Explicit program synthesis from **noisy data**.
- The program **explains** the logical relations between inputs and outputs.
- The program **predicts** the output




| | *Procedure is implicit* | *Procedure is explicit* |
|---|---|---|
| *Symbolic search* | | Symbolic Program Synthesis |
| *Optimisation procedure* | Neural Program Induction | **Neural Program Synthesis** |

MANCHESTER 1824
The University of Manchester

# Recap

## Learning Logic Programs

An **ILP problem** is a tuple $(\mathcal{B}, \mathcal{P}, \mathcal{N})$ of ground atoms, where:

- $\mathcal{B}$ is a set of background assumptions, a set of ground atoms[8].
- $\mathcal{P}$ is a set of positive instances - examples taken from the extension of the target predicate to be learned
- $\mathcal{N}$ is a set of negative instances - examples taken outside the extension of the target predicate

Given an ILP problem $(\mathcal{B}, \mathcal{P}, \mathcal{N})$, a **solution** is a set $R$ of definite clauses such that

- $\mathcal{B}, R \models \gamma$ for all $\gamma \in \mathcal{P}$
- $\mathcal{B}, R \not\models \gamma$ for all $\gamma \in \mathcal{N}$

MANCHESTER 1824
The University of Manchester

# Recap

## Learning Logic Programs

$$\mathcal{B} = \{zero(0), succ(0,1), succ(1,2), succ(2,3), ...\}$$

$$\begin{aligned} \mathcal{P} &= \{\text{even}(0), \text{even}(2), \text{even}(4), \text{even}(6), ...\} \\ \mathcal{N} &= \{\text{even}(1), \text{even}(3), \text{even}(5), \text{even}(7), ...\} \end{aligned}$$

$$\begin{aligned} even(X) &\leftarrow zero(X) \\ even(X) &\leftarrow even(Y), succ2(Y,X) \\ succ2(X,Y) &\leftarrow succ(X,Z), succ(Z,Y) \end{aligned}$$

MANCHESTER 1824
The University of Manchester

# Converting ILP to SAT

- Define a rule template $\tau$ as a way of defining a set of clauses.
- Define a program template as a set of rule templates.
- For each rule template $\tau$, generate the set $cl(\tau)$ of all clauses that satisfy the template.
- Introduce a boolean flag for each generated clause, indicating whether it is “on” or “off”.
- Now the induction problem has been transformed into a satisfiability problem: find an assignment to the flags such that the set of clauses that are “on” together entail the positive examples and do not entail the negative examples.

MANCHESTER 1824
The University of Manchester

# Restricting the Set of Rules

Restrictions made without loss of generality:

- All clauses have exactly two atoms in the body
- Each predicate is defined by exactly two clauses

Other restrictions:

- Constants in rules are not allowed
- Nullary, unary, and binary predicates

MANCHESTER 1824
The University of Manchester

# ∂ILP

∂ILP uses a **differentiable** model of **forward chaining inference.**

A valuation is a vector in $[0,1]^n$ that maps each of n ground atoms to [0,1]

A valuation represents how likely it is that each of the ground atoms is true.

| $G$ | $\mathbf{a}_0$ |
|---|---|
| $p(a)$ | 0.0 |
| $p(b)$ | 0.0 |
| $q(a)$ | 0.1 |
| $q(b)$ | 0.3 |
| $\bot$ | 0.0 |

MANCHESTER 1824
The University of Manchester

∂ILP

Each clause c is compiled into a function on valuations:

$$F_c : [0,1]^n \to [0,1]^n$$

Given the following clause:

$$p(X) \leftarrow q(X)$$

| $G$ | $\mathbf{a}_0$ | $\mathcal{F}_c(\mathbf{a_0})$ |
|---|---|---|
| $p(a)$ | 0.0 | 0.1 |
| $p(b)$ | 0.0 | 0.3 |
| $q(a)$ | 0.1 | 0.0 |
| $q(b)$ | 0.3 | 0.0 |
| $\bot$ | 0.0 | 0.0 |

MANCHESTER 1824
The University of Manchester

# ∂ILP

- Axioms = background knowledge
- Target atoms = positive (label = 1) and negative (label = 0) examples
- The initial valuation assigns 1 to each atom in the Background Knowledge, 0 otherwise
- A set of clauses is generated from a program template and a language
- The goal is to learn the right clause weights to produce the desired conclusions

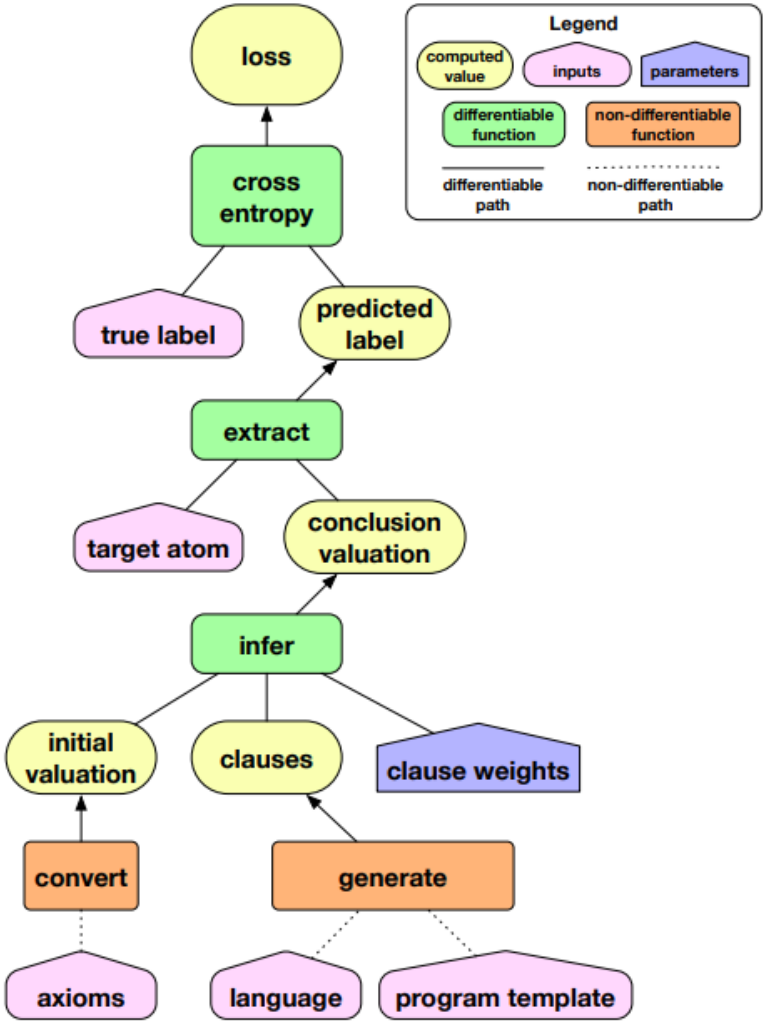



Figure 1: The ∂ILP Architecture.

MANCHESTER 1824
The University of Manchester

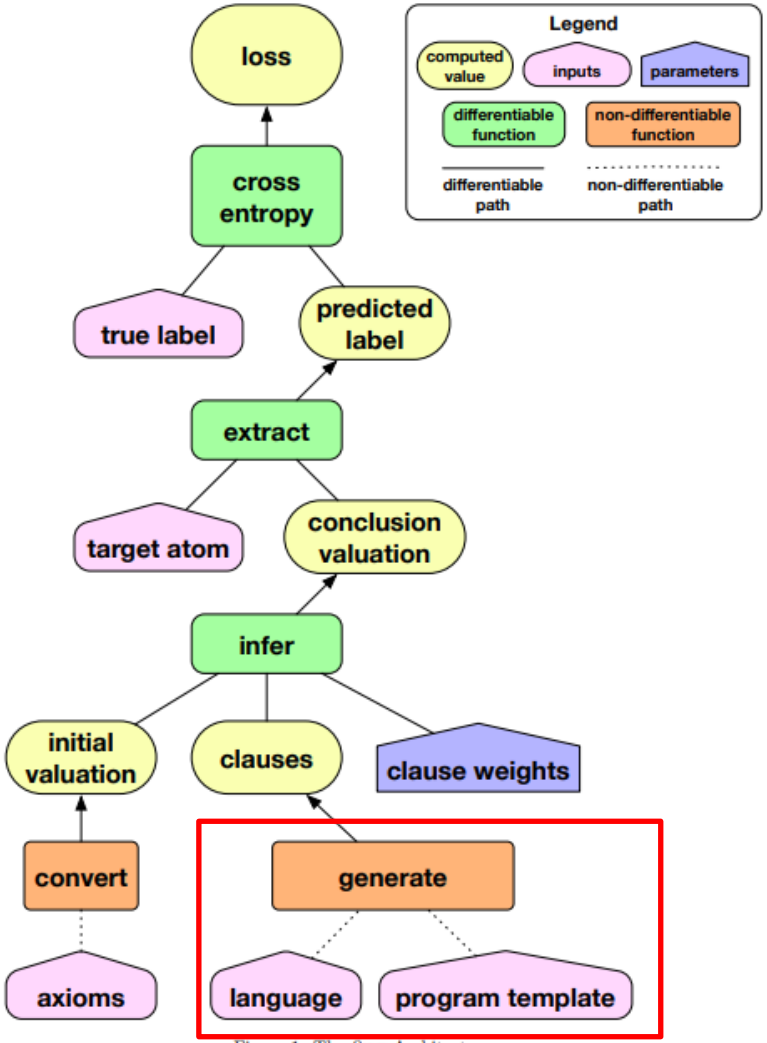



Figure 1: The ∂ILP Architecture.

MANCHESTER 1824
The University of Manchester

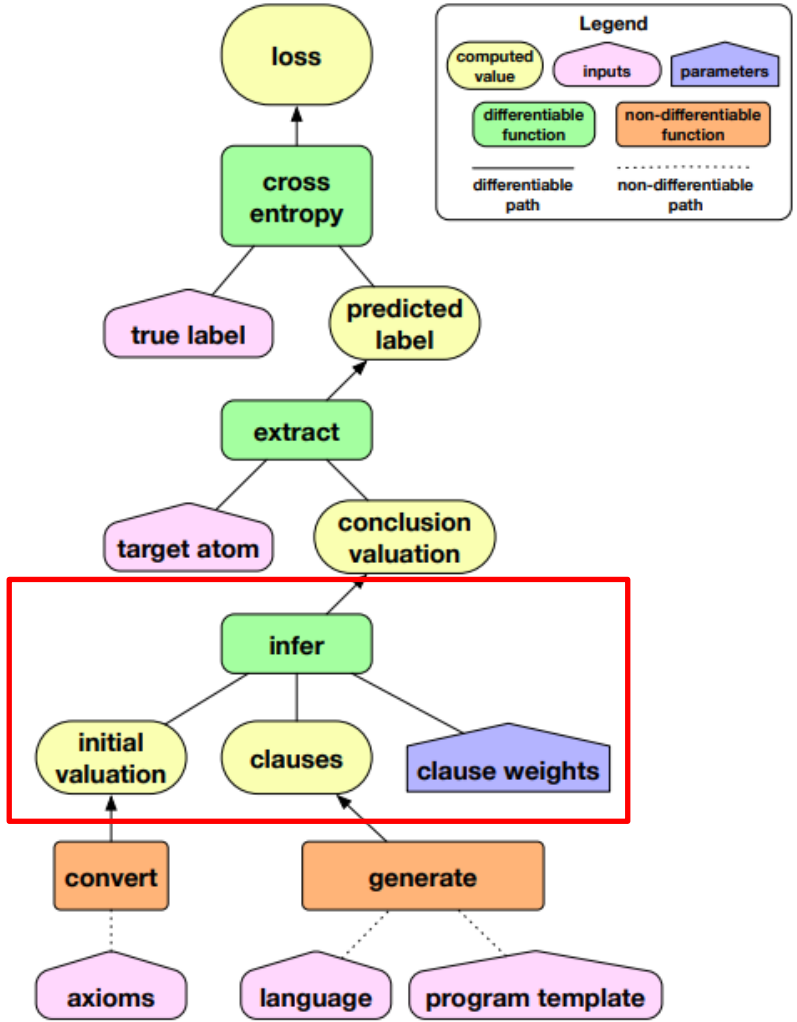



Figure 1: The $\partial$ILP Architecture.

MANCHESTER 1824
The University of Manchester

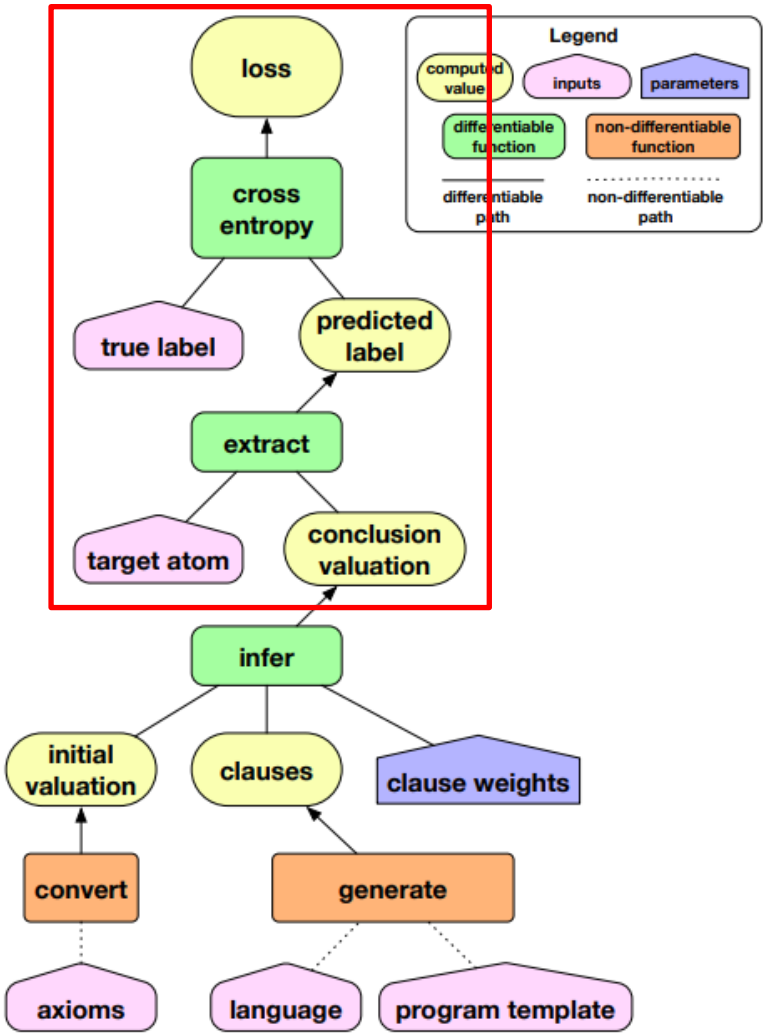



Figure 1: The $\partial$ILP Architecture.

# ∂ILP Inference

$$r(X, Y) \leftarrow p(X, Z), q(Z, Y)$$

Assume that each clause has two atoms in the body. Calculate, for each ground atom, the pairs of ground atoms that contribute to its truth:

$$\begin{aligned}
r(a,a) &: \{(p(a,a), q(a,a)),\ (p(a,b), q(b,a))\} \\
r(a,b) &: \{(p(a,a), q(a,b)),\ (p(a,b), q(b,b))\} \\
r(b,a) &: \{(p(b,a), q(a,a)),\ (p(b,b), q(b,a))\} \\
r(b,b) &: \{(p(b,a), q(a,b)),\ (p(b,b), q(b,b))\}
\end{aligned}$$

convert the pairs of atoms into pairs of indices:

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 0 | $\bot$ | {} |
| 1 | $p(a,a)$ | {} |
| 2 | $p(a,b)$ | {} |
| 3 | $p(b,a)$ | {} |
| 4 | $p(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 5 | $q(a,a)$ | {} |
| 6 | $q(a,b)$ | {} |
| 7 | $q(b,a)$ | {} |
| 8 | $q(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 9 | $r(a,a)$ | {(1,5), (2, 7)} |
| 10 | $r(a,b)$ | {(1, 6), (2, 8)} |
| 11 | $r(b,a)$ | {(3, 5), (4, 7)} |
| 12 | $r(b,b)$ | {(3, 6), (4, 8)} |

# ∂ILP Inference

convert:

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 0 | $\bot$ | {} |
| 1 | $p(a,a)$ | {} |
| 2 | $p(a,b)$ | {} |
| 3 | $p(b,a)$ | {} |
| 4 | $p(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 5 | $q(a,a)$ | {} |
| 6 | $q(a,b)$ | {} |
| 7 | $q(b,a)$ | {} |
| 8 | $q(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 9 | $r(a,a)$ | {(1,5), (2, 7)} |
| 10 | $r(a,b)$ | {(1, 6), (2, 8)} |
| 11 | $r(b,a)$ | {(3, 5), (4, 7)} |
| 12 | $r(b,b)$ | {(3, 6), (4, 8)} |

into a tensor of shape n * w * 2:

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 0 | $\bot$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 1 | $p(a,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 2 | $p(a,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 3 | $p(b,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 4 | $p(b,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 5 | $q(a,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 6 | $q(a,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 7 | $q(b,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 8 | $q(b,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 9 | $r(a,a)$ | $\begin{bmatrix}(1,5)\\(2,7)\end{bmatrix}$ |
| 10 | $r(a,b)$ | $\begin{bmatrix}(1,6)\\(2,8)\end{bmatrix}$ |
| 11 | $r(b,a)$ | $\begin{bmatrix}(3,5)\\(4,7)\end{bmatrix}$ |
| 12 | $r(b,b)$ | $\begin{bmatrix}(3,6)\\(4,8)\end{bmatrix}$ |

# ∂ILP Inference

Split the tensor X into two matrices of shape n * w:

$$\mathbf{X}_1 = \mathbf{X}[:, :, 0] \qquad \mathbf{X}_2 = \mathbf{X}[:, :, 1]$$

Gather up the results:

$$\mathbf{Y}_1 = \text{gather}_2(\mathbf{a}, \mathbf{X}_1) \qquad \mathbf{Y}_2 = \text{gather}_2(\mathbf{a}, \mathbf{X}_2)$$

take the element-wise product:

$$\mathbf{Z} = \mathbf{Y}_1 \odot \mathbf{Y}_2$$

Take the max across the second dimension:

$$F_c(\mathbf{a}) = \mathbf{a}' \text{ where } \mathbf{a}'[k] = \max(\mathbf{Z}[k, :])$$

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 0 | $\bot$ | {} |
| 1 | $p(a,a)$ | {} |
| 2 | $p(a,b)$ | {} |
| 3 | $p(b,a)$ | {} |
| 4 | $p(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 5 | $q(a,a)$ | {} |
| 6 | $q(a,b)$ | {} |
| 7 | $q(b,a)$ | {} |
| 8 | $q(b,b)$ | {} |

| $k$ | $\gamma_k$ | $x_k$ |
|---|---|---|
| 9 | $r(a,a)$ | {(1,5), (2, 7)} |
| 10 | $r(a,b)$ | {(1, 6), (2, 8)} |
| 11 | $r(b,a)$ | {(3, 5), (4, 7)} |
| 12 | $r(b,b)$ | {(3, 6), (4, 8)} |

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 0 | $\bot$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 1 | $p(a,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 2 | $p(a,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 3 | $p(b,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 4 | $p(b,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 5 | $q(a,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 6 | $q(a,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 7 | $q(b,a)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |
| 8 | $q(b,b)$ | $\begin{bmatrix}(0,0)\\(0,0)\end{bmatrix}$ |

| $k$ | $\gamma_k$ | $\mathbf{X}[k]$ |
|---|---|---|
| 9 | $r(a,a)$ | $\begin{bmatrix}(1,5)\\(2,7)\end{bmatrix}$ |
| 10 | $r(a,b)$ | $\begin{bmatrix}(1,6)\\(2,8)\end{bmatrix}$ |
| 11 | $r(b,a)$ | $\begin{bmatrix}(3,5)\\(4,7)\end{bmatrix}$ |
| 12 | $r(b,b)$ | $\begin{bmatrix}(3,6)\\(4,8)\end{bmatrix}$ |

MANCHESTER 1824
The University of Manchester

# ∂ILP Inference

$$r(X,Y) \leftarrow p(X,Z), q(Z,Y)$$
$$r(a,b) \leftarrow p(a,b), q(b,b)$$

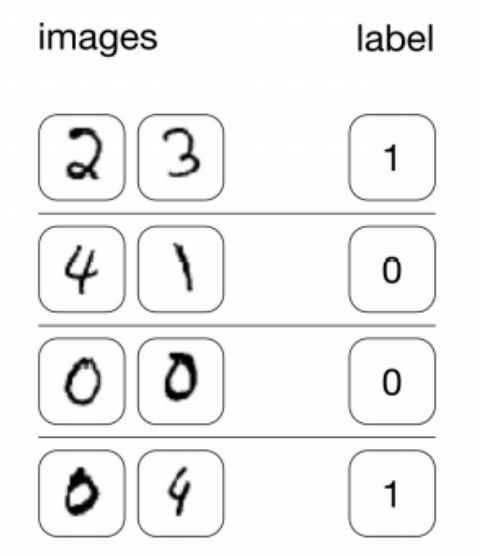
| $k$ | $\gamma_k$ | $\mathbf{a}[k]$ | $\mathbf{X}_1[k]$ | $\mathbf{X}_2[k]$ | $\mathbf{Y}_1[k]$ | $\mathbf{Y}_2[k]$ | $\mathbf{Z}[k]$ | $F_c(\mathbf{a})[k]$ |
|---|---|---|---|---|---|---|---|---|
| 0 | $\perp$ | 0.0 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 1 | $p(a,a)$ | 1.0 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 2 | $p(a,b)$ | 0.9 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 3 | $p(b,a)$ | 0.0 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 4 | $p(b,b)$ | 0.0 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 5 | $q(a,a)$ | 0.1 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 6 | $q(a,b)$ | 0.0 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 7 | $q(b,a)$ | 0.2 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 8 | $q(b,b)$ | 0.8 | [0 0] | [0 0] | [0 0] | [0 0] | [0 0] | 0.00 |
| 9 | $r(a,a)$ | 0.0 | [1 2] | [5 7] | [1.0 0.9] | [0.1 0.2] | [0.1 0.18] | 0.18 |
| 10 | $r(a,b)$ | 0.0 | [1 2] | [6 8] | [1.0 0.9] | [0 0.8] | [0 0.72] | 0.72 |
| 11 | $r(b,a)$ | 0.0 | [3 4] | [5 7] | [0 0] | [0.1 0.2] | [0 0] | 0.00 |
| 12 | $r(b,b)$ | 0.0 | [3 4] | [6 8] | [0 0] | [0 0.8] | [0 0] | 0.00 |

# ∂ILP Experiments

```
target() ← image2(X), pred1(X)

pred1(X) ← image1(Y), pred2(Y, X)

pred2(X, Y) ← succ(X, Y)

pred2(X, Y) ← pred2(Z, Y), pred2(X, Z)
```

Less than on MNIST

- Pre-trained conv-net to recognise MNIST digits.
- Convert the logits of the conv-net into a probability distribution over logical atoms.



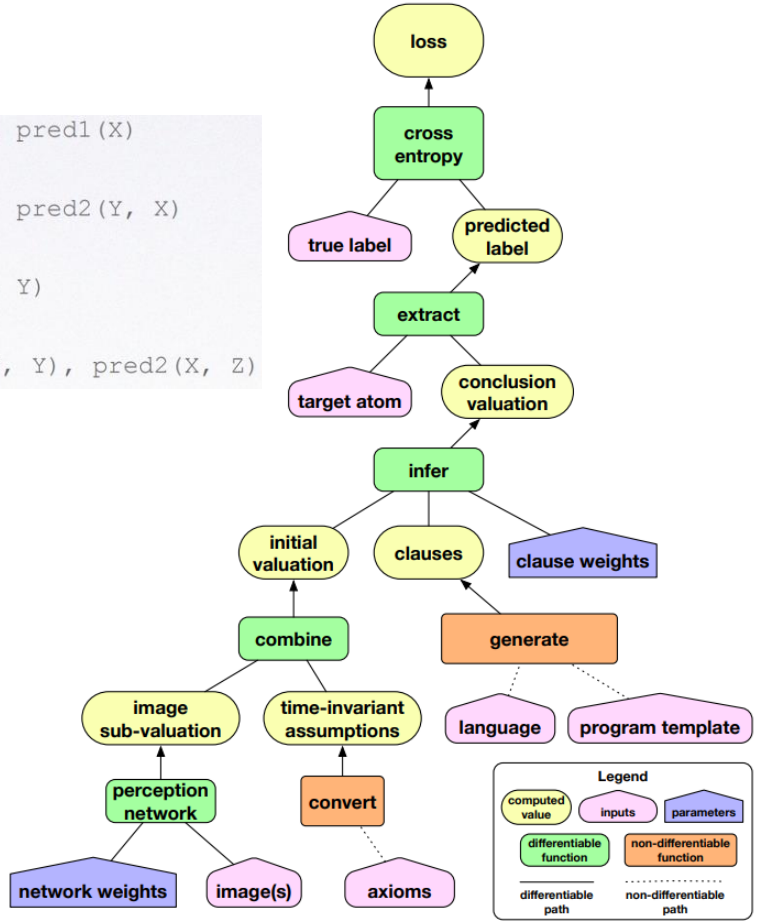




Figure 5: ∂ILP from Raw Pixel Images

MANCHESTER 1824
The University of Manchester

https://github.com/ai-systems/DILP-Core