UNIVERSITY OF EXETER

Evolution of Connections in SHRUTI Networks

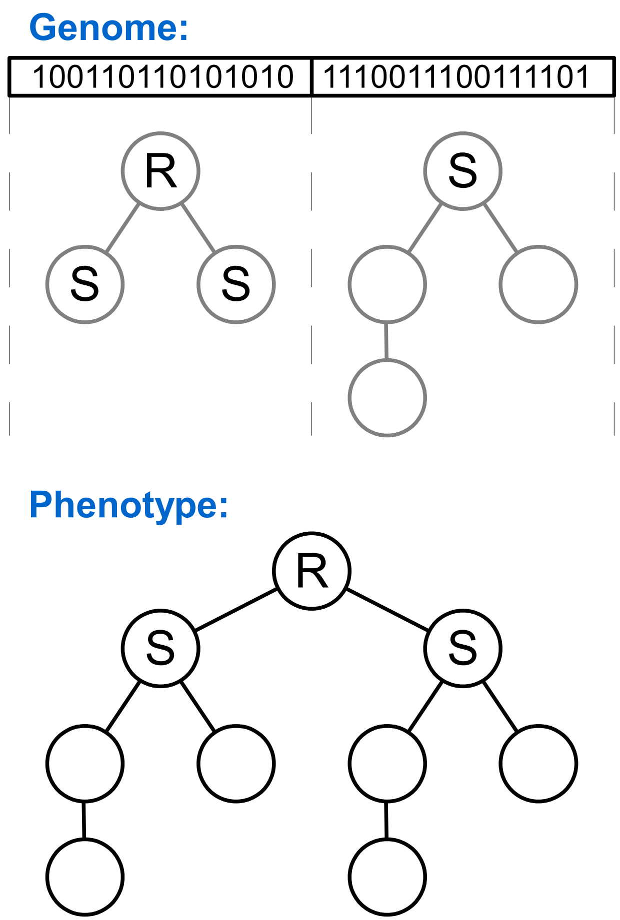
Joe Townsend, Ed Keedwell and Antony Galton

Motivation

- **Neural-Symbolic Reasoning:** Adaptable representations of logic programs in neural networks.
- Can help us understand how reasoning might be processed in the brain using real biological neurons.
- Brains, and whatever mechanisms enable them to perform reasoning, are products of evolution and development.
- Can neural-symbolic structures also be discovered through evolutionary searches?

Artificial Development

- A biologically plausible model of evolutionary computing.
- Has already been applied to the development of neural networks
- Genomes use **indirect encoding**, which like DNA, describes how the phenotype develops over time.
- Sub-structures may be discovered once in evolution and represented once in the genome, but replicated multiple times in the phenotype.

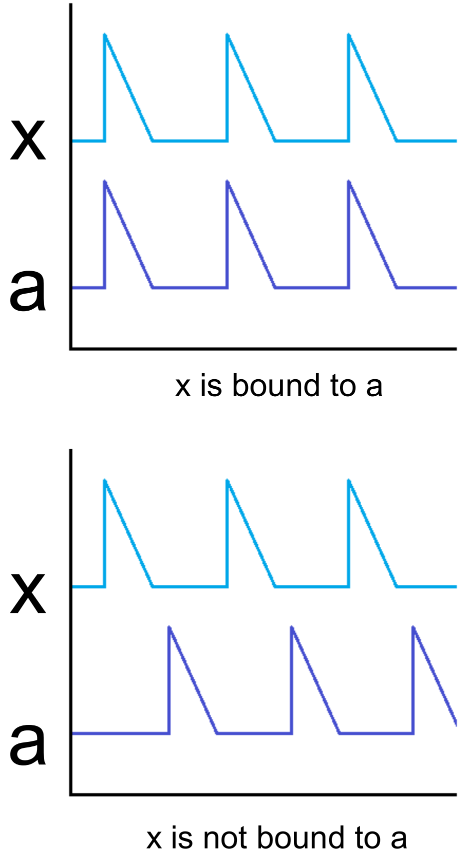


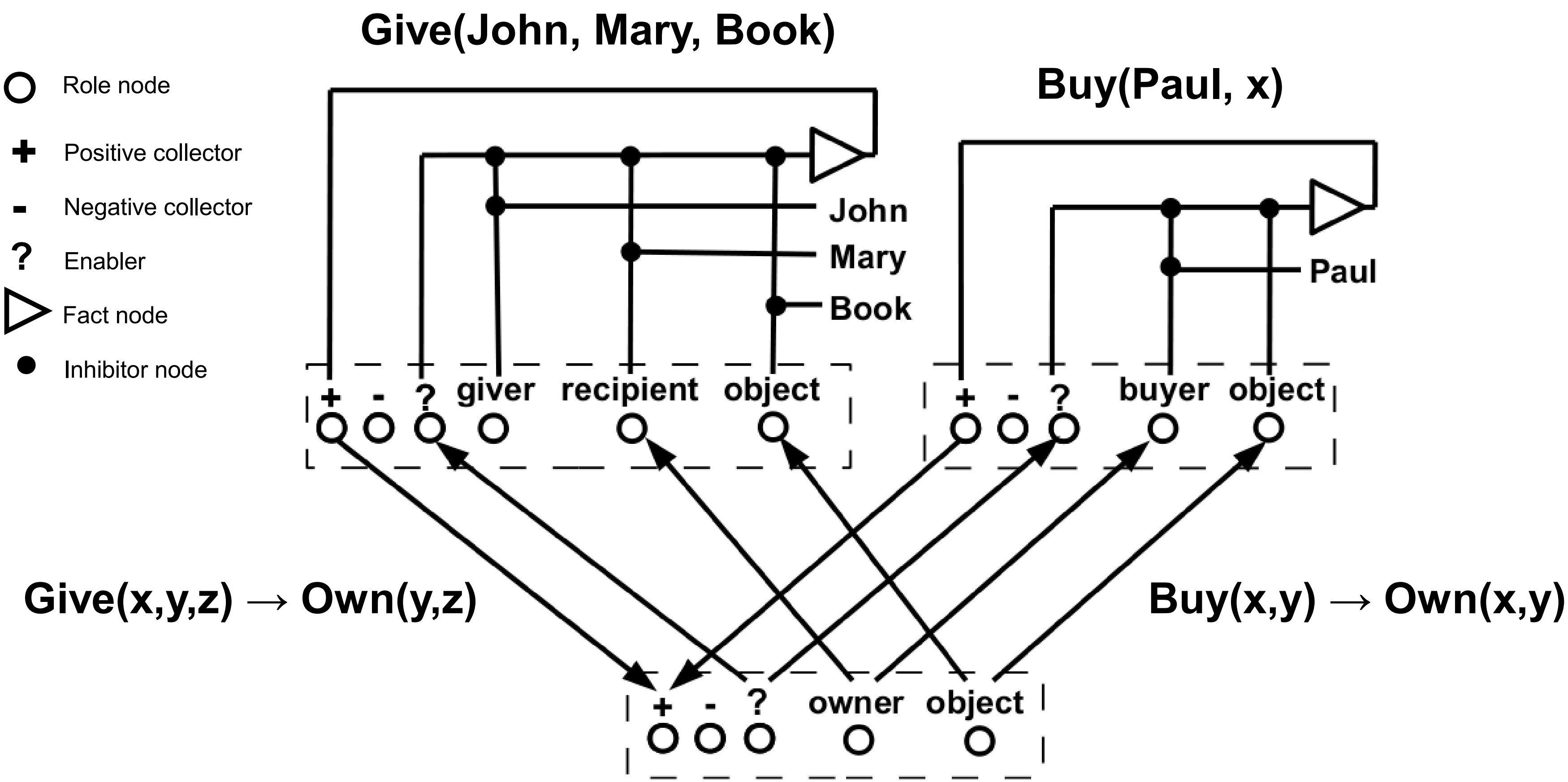
Evolving Neural-Symbolic Networks

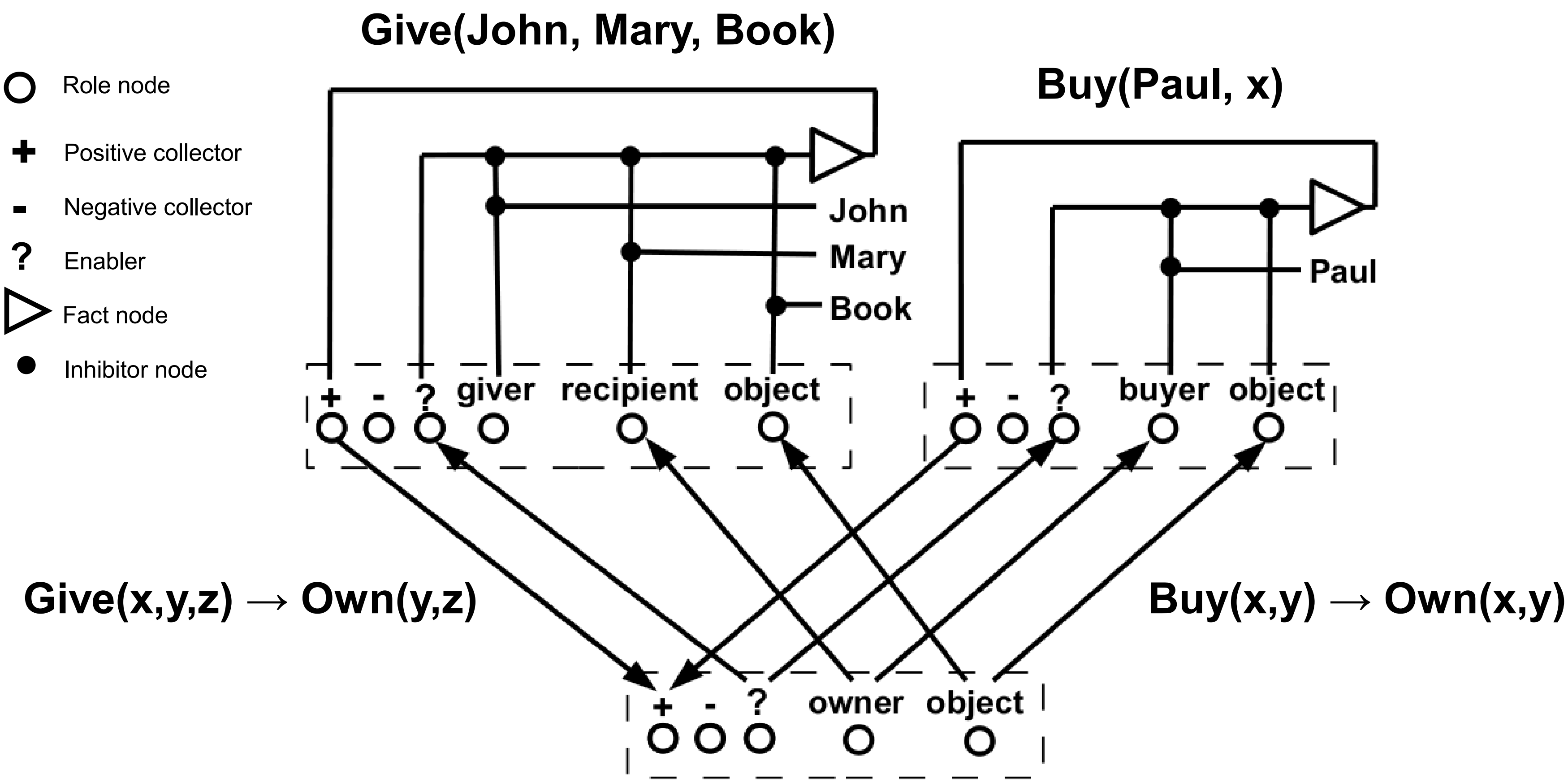
- Artificial Development been applied to the development of neural networks in general, but not specifically to neural-symbolic networks.
- These should be considered if we are to move towards evolving neural models of intelligence.
- To explore this idea, we have been attempting to rediscover and improve upon SHRUTI networks through artificial development. SHRUTI is suited for this task because:
    - Biological plausibility was one of the goals of SHRUTI
    - SHRUTI networks are constructed from smaller, repeated sub-networks, thus lending themselves well to indirect encodings.

SHRUTI [Shastri & Ajjanagadde, 1993]

- Demonstrates how predicate logic can be encoded in a network of neurons and used for reasoning.
- Uses spiking neurons, which like biological neurons, fire trains of pulses.
- Variable binding is performed by firing neurons in temporal synchrony with each other.
- Bindings can be propagated from one set of neurons to another.



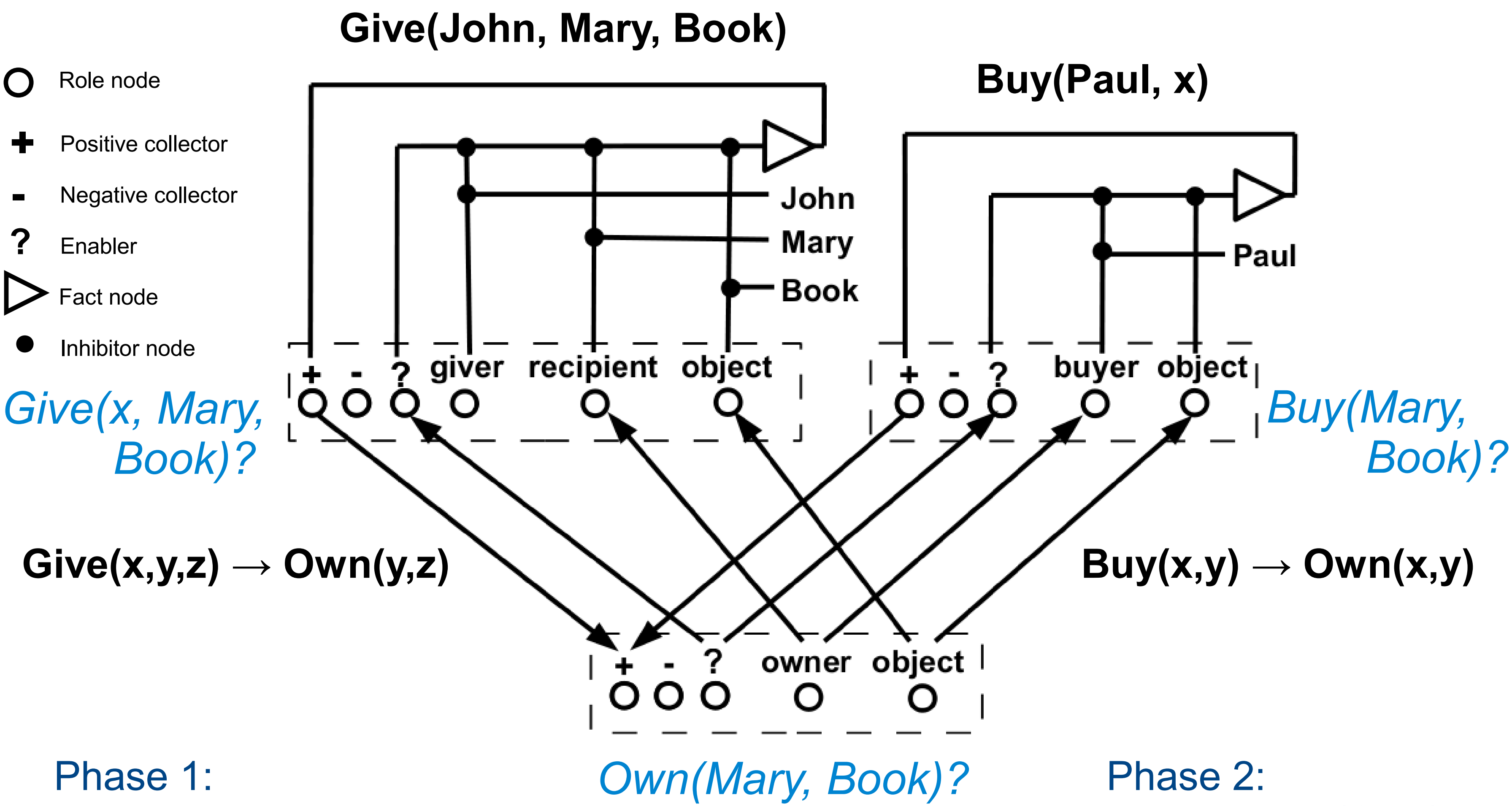

Give(John, Mary, Book)
Buy(Paul, x)
Role node
Positive collector
Negative collector
Enabler
Fact node
Inhibitor node
John
Mary
Book
Paul
+ - ? giver recipient object
+ - ? buyer object
Give(x,y,z) → Own(y,z)
Buy(x,y) → Own(x,y)
+ - ? owner object



Phase 1:
Mary, owner

Own(Mary, Book)?

Phase 2:
Book, *object(Own)*



Phase 1:
Mary, owner, Recipient, buyer

Phase 2:
Book, *object(Own), object(Give), object(Buy)*

Hebbian learning in SHRUTI

- A sequence of events in the form of predicate instances are observed.
- Observing an instance of P shortly before an instance of Q strengthens connections for the relation P → Q according to:

$$\omega_{t+1} = \omega_t + \alpha(1 - \omega_t)$$

- If P is observed but Q isn't within a fixed time window, connections for P → Q are weakened according to:

$$\omega_{t+1} = \omega_t - \alpha\, \omega_t$$

Training Data

Logic program:

- $-P(x, y, z) \rightarrow +Q(y, z)$
- $+Q(y, z) \rightarrow +S(y, z)$
- $+Q(y, z) \rightarrow -T(y, z)$
- $-R(z, x, y) \rightarrow +S(y, z)$
- $+S(y, z) \rightarrow -U(y)$

Event sequence:

1. $-P(a, b, c)$
2. $+Q(b, c)$
3. $+S(b, c), -T(b, c)$
4. $-U(b)$
5.
6.
7. $-R(f, d, e)$
8. $+S(e, f)$
9. $-U(e)$
10.
11.
12.
13. $-P(g, h, i)$
14. $+Q(h, i)$
15. $+S(h, i), -T(h, i)$
16. $-U(h)$
17.
18.
19. $-R(l, j, k)$
20. $+S(k, l)$
21. $-U(k)$
22.

Prerequisites for learning

- A logic program can be learned from a network of fully interconnected neurons.
- However, a fully interconnected network is impractical and lacks biological plausibility.
- To reduce the size of the initial network while enabling all desired relations to be learned, some pre-organisation is required.
- SHRUTI's developers argue that *“such organization could result from a genetically based developmental process” [Shastri & Wendelken, 2003]*
- Finding a genome model that uses indirect encoding to develop SHRUTI networks would support their biological plausibility.

Developing SHRUTI networks

- We have designed a genome for developing connections between neurons in a SHRUTI network.
- For each t:
  - 1. Observe events occurring at t
  - 2. Adjust existing connection weights according to the Hebbian learning algorithm.
  - 3. For each neuron pair, add or delete connection depending on rules define in genome...

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

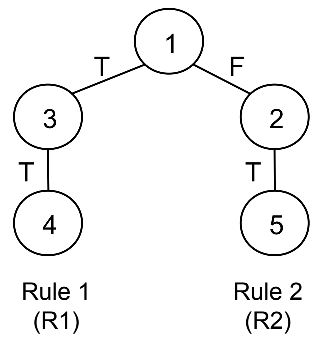
Conditions:

1. If SELF.activity > 0.5, go to 3, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. If P_INPUT.type = SELF.type, go to 4, else end.

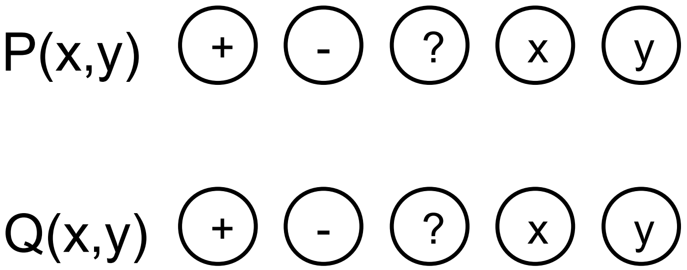
Actions:

4. Add connection with weight 0.1
5. Delete connection

Rules:



SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

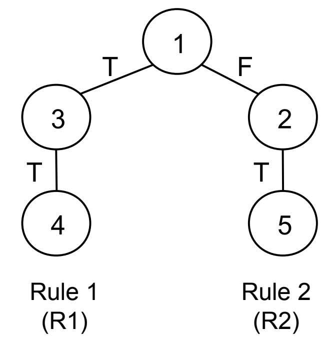
Conditions:

1. If SELF.activity > 0.5, go to 3, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. If P_INPUT.type = SELF.type, go to 4, else end.

Actions:

4. Add connection with weight 0.1
5. Delete connection

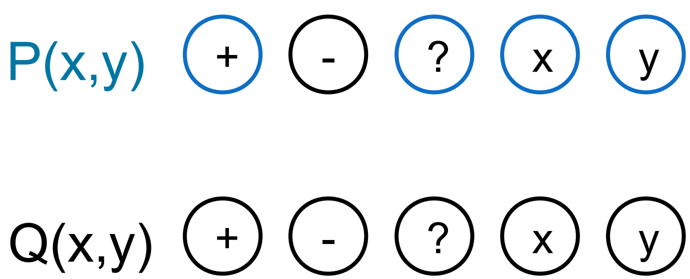
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

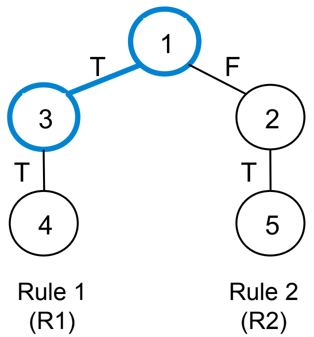
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. Add connection with weight 0.1
5. Delete connection

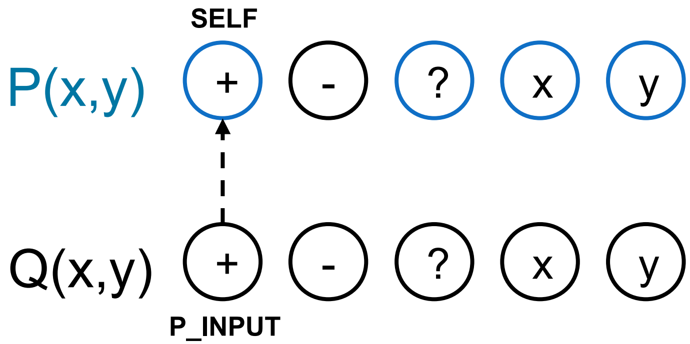
Rules:



Observation:

P(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

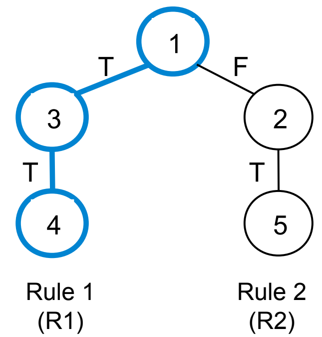
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

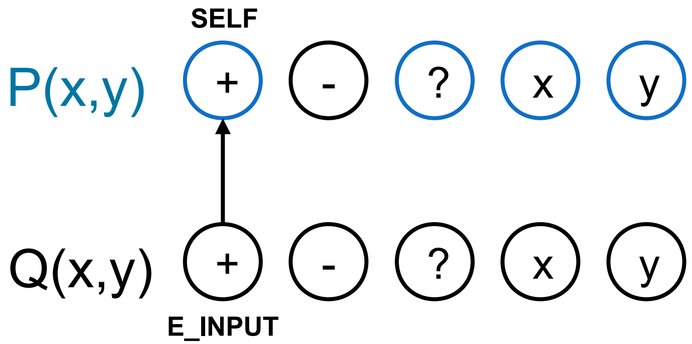
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

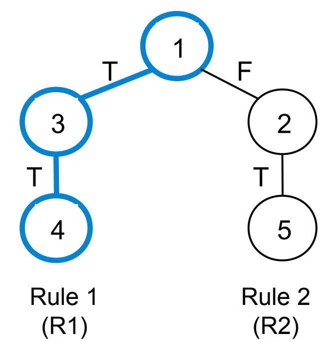
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

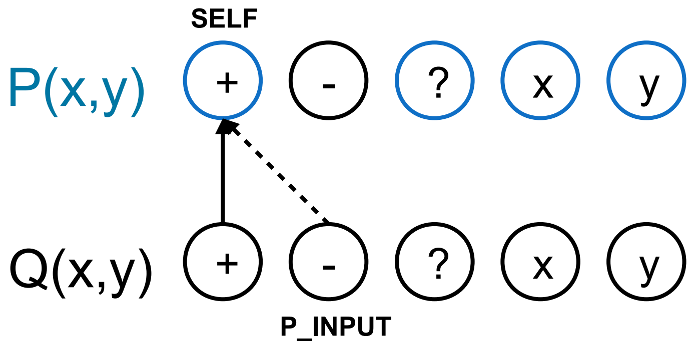
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

1. Condition					2. Condition					3. Condition					4. Action		5. Action	
SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	

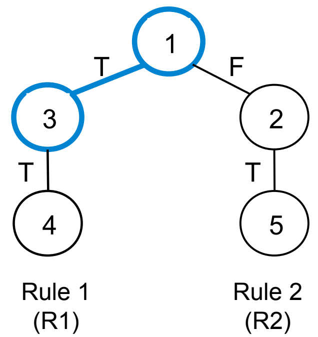
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type**, go to 4, **else end**.

Actions:

4. Add connection with weight 0.1
5. Delete connection

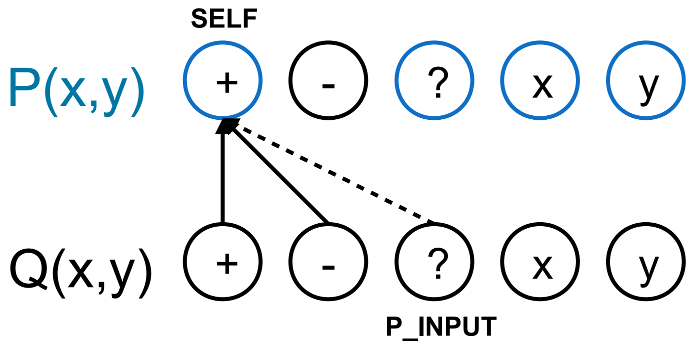
Rules:



Observation:

P(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

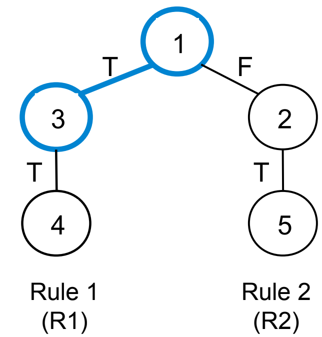
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type**, go to 4, **else end**.

Actions:

4. Add connection with weight 0.1
5. Delete connection

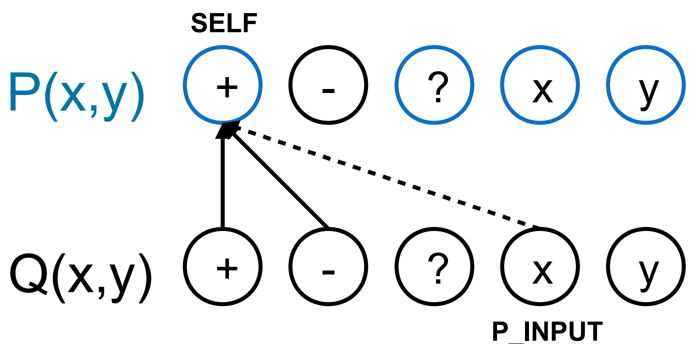
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

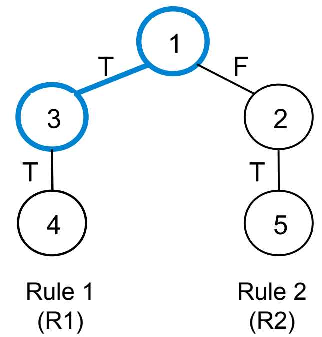
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type**, go to 4, **else end**.

Actions:

4. Add connection with weight 0.1
5. Delete connection

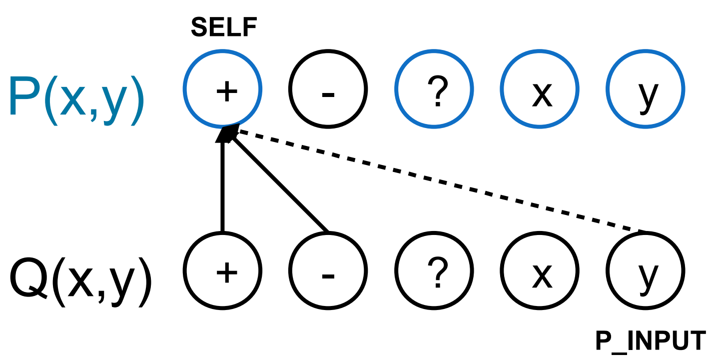
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

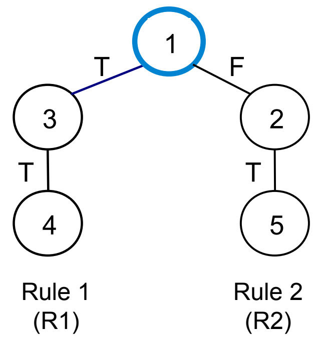
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. If P_INPUT.type = SELF.type, go to 4, else end.

Actions:

4. Add connection with weight 0.1
5. Delete connection

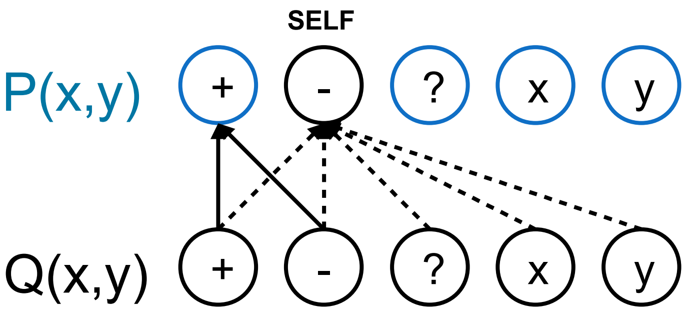
Rules:



SHRUTI network:

Observation:

P(a,b)



- ***SELF**: Neuron for which connections are being considered*
- ***P_INPUT**: Possible input*
- ***E_INPUT**: Existing input*

Developing SHRUTI networks

The Genome:

1. Condition					2. Condition					3. Condition					4. Action		5. Action	
SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	

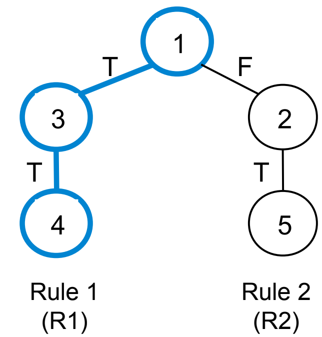
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

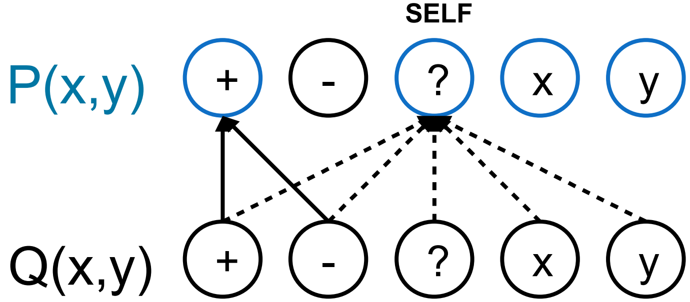
Rules:



Observation:

P(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL		
1. Condition					2. Condition					3. Condition					4. Action		5. Action		

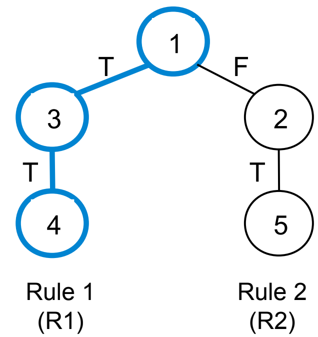
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

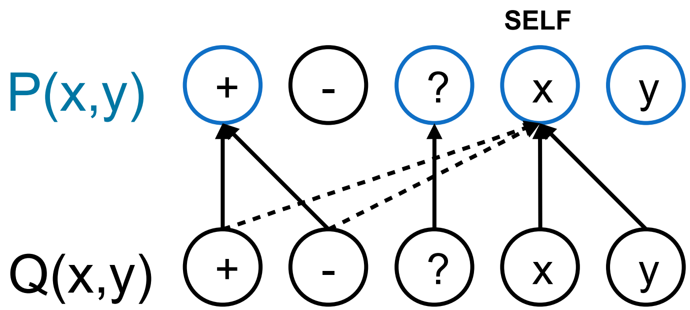
Rules:



Observation:

P(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

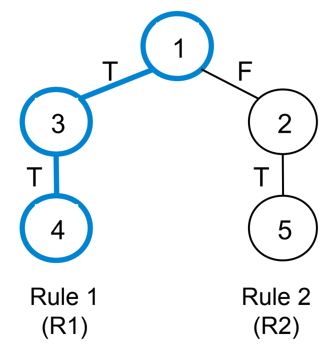
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

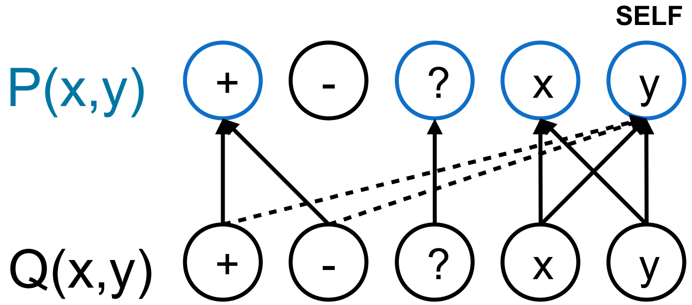
Rules:



Observation:

P(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

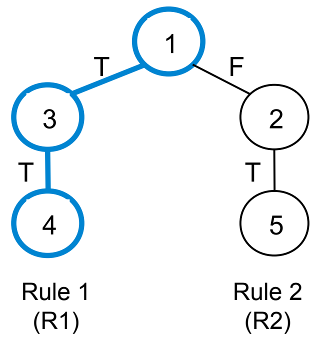
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

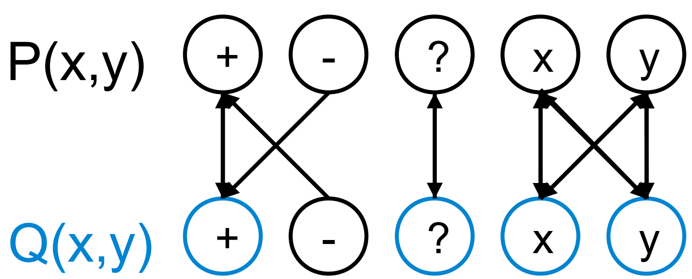
Rules:



SHRUTI network:

Observation:

Q(a,b)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

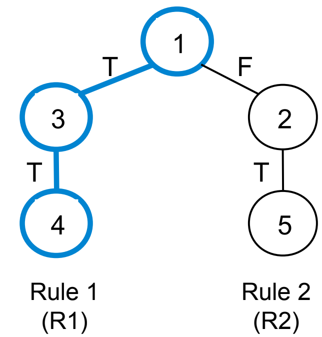
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

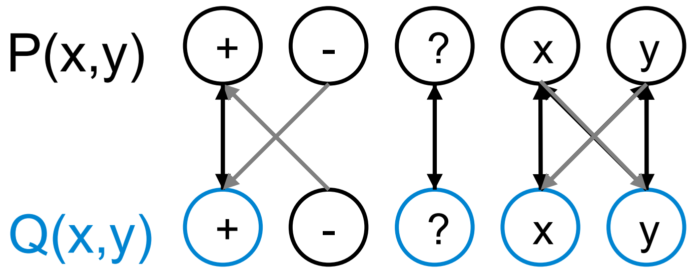
Rules:



Observation:

Q(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2

1. Condition

E_INPUT.weight	<	0.1	5	0

2. Condition

P_INPUT.type	=	SELF.type	5	0

3. Condition

ADD	0.1

4. Action

DEL	

5. Action

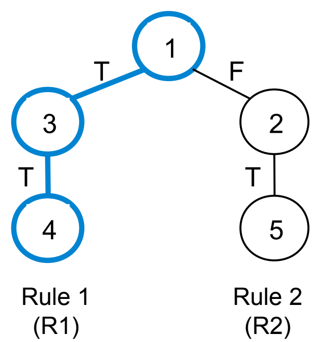
Conditions:

1. **If SELF.activity > 0.5, go to 3**, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. **If P_INPUT.type = SELF.type, go to 4**, else end.

Actions:

4. **Add connection with weight 0.1**
5. Delete connection

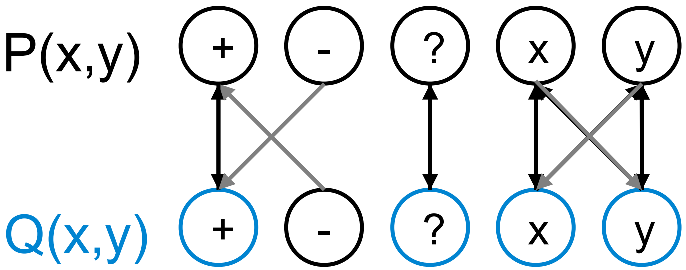
Rules:



Observation:

Q(a,b)

SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity | < | 0.5 | 3 | 2	E_INPUT.weight | < | 0.1 | 5 | 0	P_INPUT.type | = | SELF.type | 5 | 0	ADD | 0.1	DEL
1. Condition	2. Condition	3. Condition	4. Action	5. Action

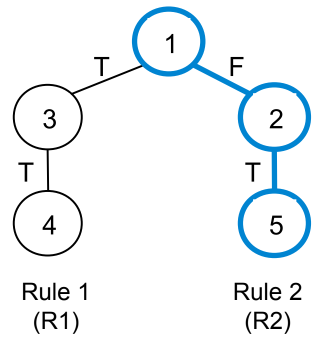
Conditions:

1. **If SELF.activity > 0.5**, go to 3, **else go to 2**
2. **If E_INPUT.weight < 0.1, go to 5**, else end.
3. If P_INPUT.type = SELF.type, go to 4, else end.

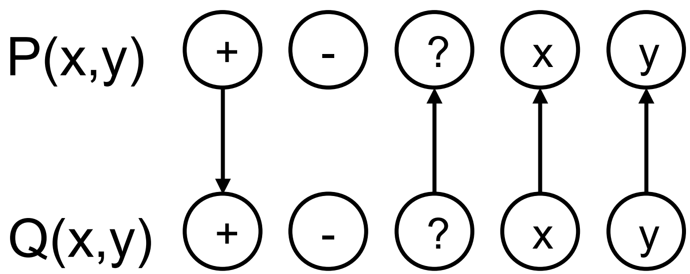
Actions:

4. Add connection with weight 0.1
5. **Delete connection**

Rules:



SHRUTI network:



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Developing SHRUTI networks

The Genome:

SELF.activity	<	0.5	3	2	E_INPUT.weight	<	0.1	5	0	P_INPUT.type	=	SELF.type	5	0	ADD	0.1	DEL	
1. Condition					2. Condition					3. Condition					4. Action		5. Action	

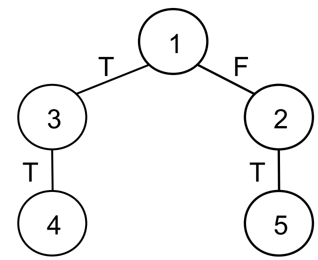
Conditions:

1. If SELF.activity > 0.5, go to 3, else go to 2
2. If E_INPUT.weight < 0.1, go to 5, else end.
3. If P_INPUT.type = SELF.type, go to 4, else end.

Actions:

4. Add connection with weight 0.1
5. Delete connection

Rules:

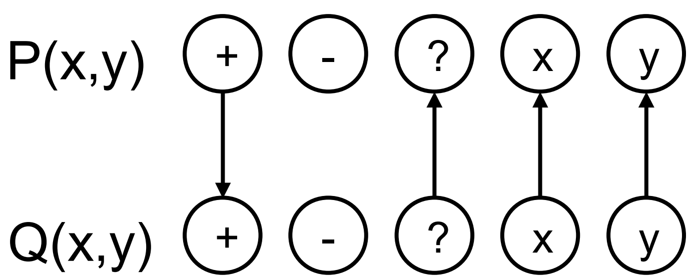


Rule 1 (R1)

Rule 2 (R2)

SHRUTI network:

P(x,y) →Q(x,y)



- ***SELF***: *Neuron for which connections are being considered*
- ***P_INPUT***: *Possible input*
- ***E_INPUT***: *Existing input*

Number of connections

Sequence	Relations	Predicates	Connections	Updates
1	2	3	22	139
2	3	4	64	719
3	4	5	86	1056
4	4	7	53	535
5	5	6	65	721
6	5	6	80	852
7	5	6	83	1064
8	6	7	80	1168
9	6	7	73	730

- Developed networks answered all test questions correctly in each case.
- Though size of genome is fixed, it can develop networks of different sizes.

Evolving the SHRUTI genome

- The genome supports the claim that the prerequisite structure required for learning relations can be realised through a model of development.
- To support this even further, we want to show that this genome can be found in a evolutionary search.

- NSGA-II
- Population size: 100
- 500 generations
- 50 trials
- 12 conditions or actions per genome
- 90% crossover rate
- 10% mutation rate

Evaluating Error

- Each network was trained on a sequence of observations in the form of predicate instances.
  - E.G. Sequence P(a,b), Q(a,b) supports the relation P(x,y) → Q(x,y)
- Networks are then presented with a set of training questions.
  - E.G. Q(a,b) - *"Is Q(a,b) true?"*
  - Answers take the form of positive and negative collector activations.

Positive Collector (+)	Negative Collector (-)	Interpretation
0	0	Unknown
1	0	True
0	1	False
1	1	Contradiction

- Error is then based on the number of questions answered incorrectly.
- Networks which always guess [0,0] (unknown) are penalised and score maximum error.

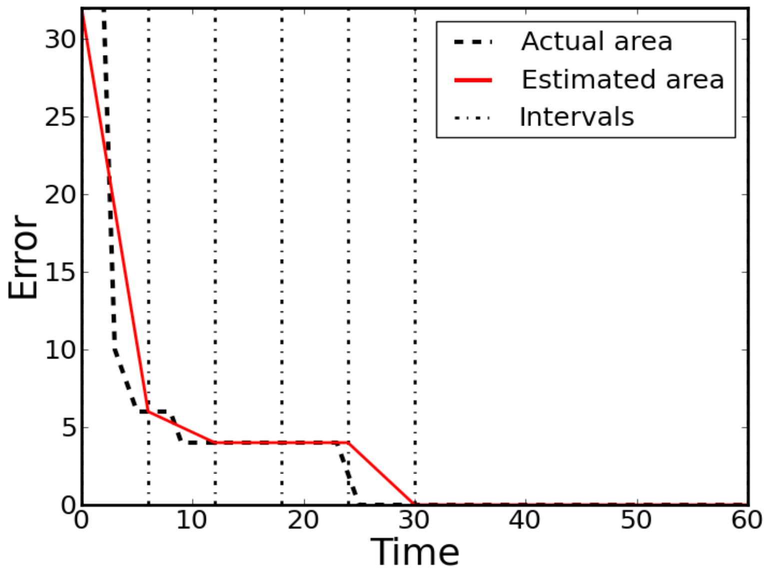
Objectives

- **Objective 1: Area beneath error-time graph**

- Algorithm converges towards genomes that produce minimal error in short amount of time.

- Error based on number of 'True or False' questions answered incorrectly. E.G. 'Is P(a,b) True?'

- Error measured at intervals during development to estimate area.



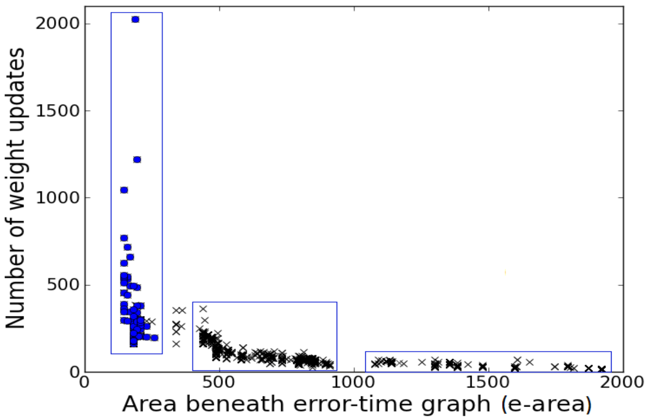
- **Objective 2: Number of weight updates**

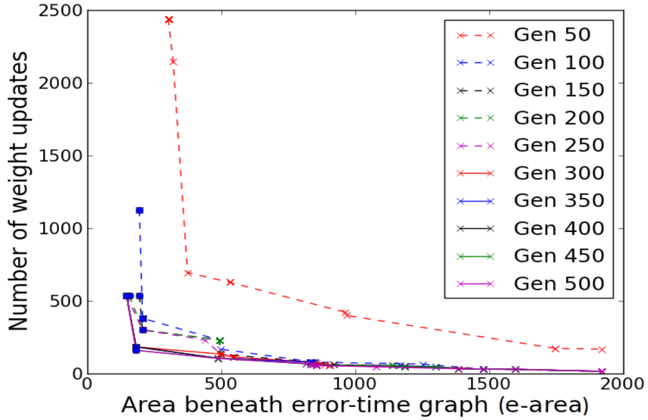
- Minimising this reduces the workload of SHRUTI's learning algorithm.

- Also reduces the number of connections, since a greater number of connections results in more weights to update during development.

Results – Training Data

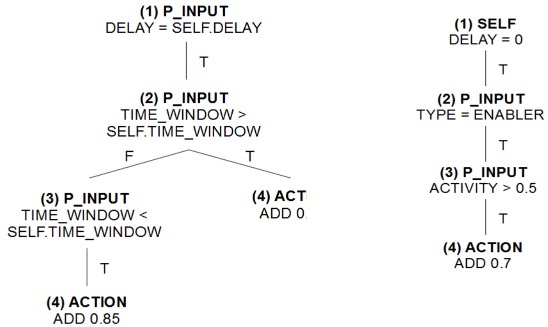
- Points marked with a blue dot show networks which yield zero-error.
- Zero-error networks could be found within the first 100 generations, but minimising the number of weight updates took longer.
- 48/50 trials yielded a total of 224 zero-error networks.
- In general, three groups of networks emerged in the final populations:
    - Zero-error networks
    - Networks which always 'guess' the same answer for different predicates
    - Networks which did not guess at all





Results – Training Data

- **First Group**
  - Zero-error networks
  - Genome only formed connections between active neurons of the same type.
- **Second Group**
  - Networks which always 'guess' the same answer depending on the predicate.
  - Only formed connections between enablers and collectors, answering questions without reference to predicate arguments.
  - Therefore relatively small number of connections and weight updates required, making members of this group difficult to dominate.
- **Third Group**
  - Maximum-error networks
  - Questions always answered 'unknown'
  - Possible with no connections, therefore also difficult to dominate.



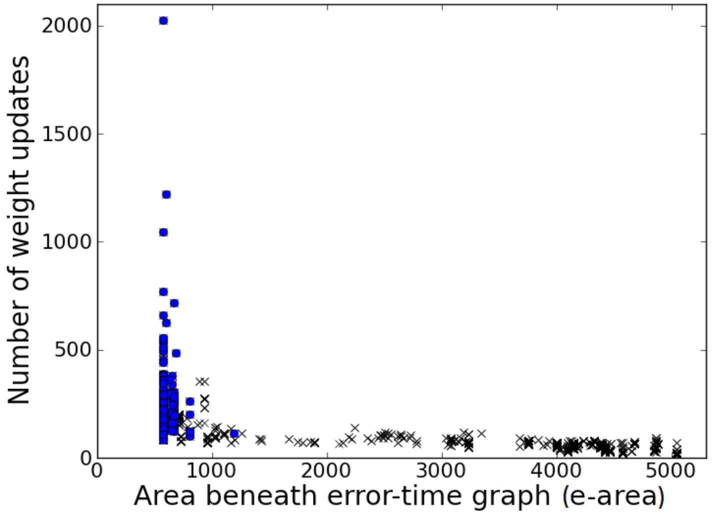
Genomes for first (left) and second (right) groups.

Question	Expected Answer	Given Answer
P(a,b)	1,0 (True)	1,0 (True)
P(c,d)	0,0 (Unknown)	1,0 (True)
Q(a,b)	0,1 (False)	0,1 (False)
Q(c,d)	0,0 (Unknown)	0,1 (False
R(a,b)	1,0 (True)	0,0 (Unknown)
R(c,d)	0,0 (Unknown)	0,0 (Unknown)

Example answering strategy for second group

Results – Test Questions

- The evolved genomes were presented with a set of test questions.
- In general, most networks performed as well on test questions as they did on training questions.
- Similar pareto fronts obtained for all trials.



Test data - Other event sequences

- Genomes from all trials were tested on different event sequences and test questions corresponding to the logic programs they support.
- In most cases, at least 90% of zero-error genomes could produce zero-error networks for the unseen data.
- The results show that the evolved SHRUTI genomes adapt well to unseen data.
- This is because genomes evolve to represent a generic relation between predicates, and not the logic program as a whole.
- The structure is discovered once, encoded once, and expressed multiple times.

Sequence	Trials with zero-error networks	# Zero-error networks
Original	*48*	*224*
1	48	218 (97%)
2	48	220 (98%)
3	25	62 (28%)
4	43	176 (78%)
5	48	200 (89%)
6	48	224 (100%)
7	48	222 (99%)
8	47	213 (95%)

Conclusions

• Artificial Development may help us in our search for neural models of intelligence. The ability to reason is one important skill which such models should exhibit.

• A step in this direction has been made by showing that a scalable genome for developing connections between neurons in SHRUTI networks can be found through evolution.

• Genomes are adaptable and scalable, owing to the fact that the genome evolves such that it learns to represent a relation between the two predicates rather than logic programs as a whole.

• This supports the idea that the pre-organisation required to learn SHRUTI relations with minimum number of connections is biologically plausible through a model of genetic development.

Future Work

- Genomes for developing other SHRUTI structures not yet covered include conjunctive relations, facts and type hierarchies.
- Currently working with genome for relations and facts, but these require much more complex representations, and attempts to find them in an evolutionary search are as of yet unsuccessful.
- Discovery of alternatives to SHRUTI.

Thank you!

Joe Townsend, University of Exeter

jt231@ex.ac.uk