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)

Give(x,y,z) → Own(y,z)

Buy(x,y) → Own(x,y)
Give(John, Mary, Book)

Buy(Paul, x)

Give(x,y,z) → Own(y,z)

Buy(x,y) → Own(x,y)

Phase 1:
Mary, owner

Own(Mary, Book)?

Phase 2:
Book, object(Own)
Give(John, Mary, Book)

Buy(Paul, x)

Give(x, Mary, Book) → Own(y, z)

Buy(x, y) → Own(x, y)

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:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &lt; 0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Actions:

4. Add connection with weight 0.1
5. Delete connection

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:

1. Rule 1 (R1)
2. Rule 2 (R2)

SHRUTI network:

\[ P(x,y) + - ? x y \]

\[ Q(x,y) + - ? x y \]

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

The Genome:

<table>
<thead>
<tr>
<th>SELF.activity</th>
<th>E_INPUT.weight</th>
<th>P_INPUT.type</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 0.5</td>
<td>&lt; 0.1</td>
<td>= SELF.type</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

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:

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

Observation:

P(a,b)

P(x,y) + - ? x y

Q(x,y) + - ? x y

SHRUTI network:
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. If SELF.activity &gt; 0.5, go to 3, else go to 2</td>
<td>4. Add connection with weight 0.1</td>
</tr>
<tr>
<td>2. If E_INPUT.weight &lt; 0.1, go to 5, else end.</td>
<td>5. Delete connection</td>
</tr>
<tr>
<td>3. If P_INPUT.type = SELF.type, go to 4, else end.</td>
<td></td>
</tr>
</tbody>
</table>

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

The SHRUTI network:

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

Observation:

\[ P(a,b) \]
\[ P(x,y) \]
\[ Q(x,y) \]
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>SELF.activity</th>
<th>&lt;</th>
<th>0.5</th>
<th>3</th>
<th>2</th>
<th>E_INPUT.weight</th>
<th>&lt;</th>
<th>0.1</th>
<th>5</th>
<th>0</th>
<th>P_INPUT.type</th>
<th>=</th>
<th>SELF.type</th>
<th>5</th>
<th>0</th>
<th>ADD</th>
<th>0.1</th>
<th>DEL</th>
</tr>
</thead>
</table>

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)

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:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &lt; 0.5</td>
<td>3</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type</td>
<td>0</td>
</tr>
</tbody>
</table>

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:
1. Rule 1 (R1)
2. Rule 2 (R2)

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:

<table>
<thead>
<tr>
<th>SELF.activity &lt; 0.5</th>
<th>3</th>
<th>2</th>
<th>E_INPUT.weight &lt; 0.1</th>
<th>5</th>
<th>0</th>
<th>P_INPUT.type = SELF.type</th>
<th>5</th>
<th>0</th>
<th>ADD 0.1</th>
<th>DEL</th>
</tr>
</thead>
</table>

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:
1. Rule 1
2. Rule 2

Observation:
P(a,b) P(x,y) Q(x,y)

SHRUTI network:

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

The Genome:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &gt; 0.5, go to 3, else go to 2</td>
<td>Add connection with weight 0.1</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1, go to 5, else end.</td>
<td>Delete connection</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type, go to 4, else end.</td>
<td></td>
</tr>
</tbody>
</table>

**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

**Observation:**

- **P(a,b)**
- **P(x,y)**
- **Q(x,y)**

**SHRUTI network:**

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

**Rules:**
- Rule 1 (R1)
- Rule 2 (R2)

---

www.exeter.ac.uk
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Condition</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &lt; 0.5</td>
<td>E_INPUT.weight &lt; 0.1</td>
<td>P_INPUT.type = SELF.type</td>
<td>ADD 0.1 DEL</td>
</tr>
</tbody>
</table>

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)**: T
- **Rule 2 (R2)**: F

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

Observation:
- \( P(a,b) \)
- \( P(x,y) \)
- \( Q(x,y) \)
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &gt; 0.5, go to 3, else go to 2</td>
<td>1. Condition</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1, go to 5, else end.</td>
<td>2. Condition</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type, go to 4, else end.</td>
<td>3. Condition</td>
</tr>
<tr>
<td>Add connection with weight 0.1</td>
<td>4. Action</td>
</tr>
<tr>
<td>Delete connection</td>
<td>5. Action</td>
</tr>
</tbody>
</table>

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:
1. Add connection with weight 0.1
2. Delete connection

Observation:

\[ P(a,b) \]

\[ P(x,y) \]

\[ Q(x,y) \]

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
- E_INPUT.weight < 0.1
- P_INPUT.type = SELF.type
- 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

The SHRUTI network:

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

Observation:

P(a,b)
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &gt; 0.5, go to 3</td>
<td>Add connection with weight 0.1</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1, go to 5</td>
<td>Delete connection</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type, go to 4</td>
<td></td>
</tr>
</tbody>
</table>

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

Observation:
\[ P(a,b) \]
\[ P(x,y) \]
\[ Q(x,y) \]

SHRUTI network:

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

The Genome:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Condition</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &lt; 0.5</td>
<td>E_INPUT.weight &lt; 0.1</td>
<td>P_INPUT.type = SELF.type</td>
<td>ADD 0.1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>DEL</td>
</tr>
</tbody>
</table>

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:
1. Rule 1 (R1)
2. Rule 2 (R2)

Observation:

P(a,b)

Q(x,y)

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
- E_INPUT.weight < 0.1
- P_INPUT.type = SELF.type
- 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:
1. Rule 1 (R1)
2. Rule 2 (R2)

Observation:
Q(a,b) P(x,y) Q(x,y)

SHRUTI network:

- SELF: Neuron for which connections are being considered
- P_INPUT: Possible input
- E_INPUT: Existing input
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &gt; 0.5</td>
<td>Add connection with weight 0.1</td>
</tr>
<tr>
<td>E_INPUT.weight &lt; 0.1</td>
<td>Delete connection</td>
</tr>
<tr>
<td>P_INPUT.type = SELF.type</td>
<td></td>
</tr>
</tbody>
</table>

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

Observation:

Q(a,b)  P(x,y)  Q(x,y)

SHRUTI network:

- SELF: Neuron for which connections are being considered
- P_INPUT: Possible input
- E_INPUT: Existing input
Developing SHRUTI networks

The Genome:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Condition</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF.activity &lt; 0.5</td>
<td>E_INPUT.weight &lt; 0.1</td>
<td>P_INPUT.type = SELF.type</td>
<td>ADD 0.1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

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

The Genome:

Rules:
- Rule 1 (R1):
  - 1. True
  - 2. True
  - 3. True
  - 4. False
  - 5. False

Rule 2 (R2):
- 1. False
- 2. False
- 3. False
- 4. True
- 5. True

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:

<table>
<thead>
<tr>
<th>SELF.activity</th>
<th>E_INPUT.weight</th>
<th>P_INPUT.type</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 0.5</td>
<td>&lt; 0.1</td>
<td>= SELF.type</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

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

SHRUTI network:

P(x,y) + - ? x y
Q(x,y) + - ? x y

• SELF: Neuron for which connections are being considered
• P_INPUT: Possible input
• E_INPUT: Existing input
Developing SHRUTI networks

**The Genome:**

<table>
<thead>
<tr>
<th>SELF.activity</th>
<th>&lt;</th>
<th>0.5</th>
<th>3</th>
<th>2</th>
<th>E_INPUT.weight</th>
<th>&lt;</th>
<th>0.1</th>
<th>5</th>
<th>0</th>
<th>PINPUT.type</th>
<th>=</th>
<th>SELF.type</th>
<th>5</th>
<th>0</th>
<th>ADD</th>
<th>0.1</th>
<th>DEL</th>
</tr>
</thead>
</table>

**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:**

\[ P(x,y) \rightarrow Q(x,y) \]

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

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Relations</th>
<th>Predicates</th>
<th>Connections</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>22</td>
<td>139</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>64</td>
<td>719</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>86</td>
<td>1056</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>7</td>
<td>53</td>
<td>535</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>65</td>
<td>721</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>80</td>
<td>852</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>83</td>
<td>1064</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>7</td>
<td>80</td>
<td>1168</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>7</td>
<td>73</td>
<td>730</td>
</tr>
</tbody>
</table>

- 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 an 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.

<table>
<thead>
<tr>
<th>Positive Collector (+)</th>
<th>Negative Collector (-)</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Unknown</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>True</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>False</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Contradiction</td>
</tr>
</tbody>
</table>

• 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.
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.

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