Adaptive Neuro-Symbolic Network Agent

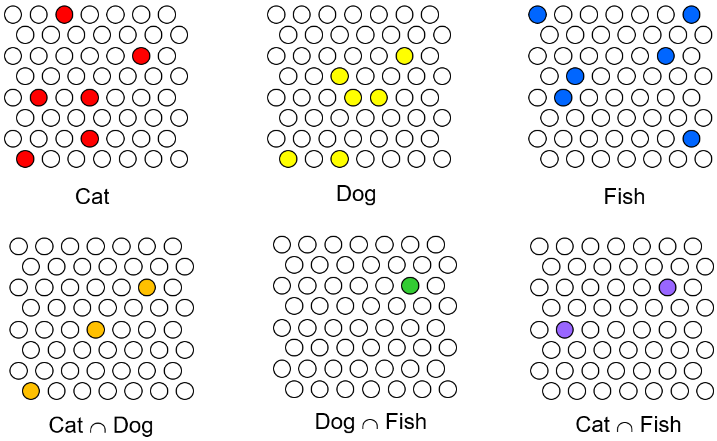
Patrick Hammer
patrick.hammer@temple.edu

ANSNA

- Sensorimotor System
- NARS heritage, using NAL-1/7/8
- Uses Sparse Distributed Representations, not Terms
- Event driven control ideas adapted from:
  Tony Lofthouse's Adaptive Logic And Neural Network
  (https://cis.temple.edu/tagit/events/papers/Lofthouse.pdf **)**
- Uses Procedure Learning ideas from OpenNARS
- Can efficiently process sensor information

Knowledge Representation

- SDR's, supporting intersection **SDRInt(a,b) = a & b**



- Also **SDRUnion(a,b) = a | b** and tuple formation:
- **SDRTuple(a,b) =** $\mathbf{\Pi_S(a) \oplus \Pi_P(b)}$ used for sequences

(Picture from http://www.sparsey.com)

Event

Composed of:

- SDR
- Type: Belief or Goal
- NAL Truth Value: Frequency and Confidence
- Evidental Base: Input ID's, to count evidence once
- Occurrence Time

Concept

Composed of:

- SDR
- Usage (useCount and lastUsed)
- Belief and goal event
- Predictive links

also:

- created for new event when novel enough
- when an event matches it, Usage gets updated, and Belief/Goal event revises with matched event:

Matching event e to concept C

Match event **e** to concept with highest truth expectation of
T_match = Truth of **e --> C**: (winner takes all / Choice)

- Positive evidence **E_p = {i | C_i=e_i=1}**
- Negative evidence **E_n = {i | e_i=1 ∧ C_i=0}**
- Total evidence **E_total = E_p ∪ E_n**

- Match frequency **f_match = |E_p| / |E_total|**
- Match conf. **c_match = |E_total| / (|E_total| + 1)**
- Match truth **T_match = (f_match, c_match)**

- **SDR_Inheritance(S,P) = Deduction(T_match, T_E)**
  ... Essentially Revision plus Deduction

Predictive Links

Predictive NAL-8 statement **(&/,a,op) =/> b** composed of:

- Source Concept SDR **a**
- Target Concept SDR **b**
- Operation **op** (can be **None**)
- NAL Truth Value
- Occurrence time offset
- Occurrence time variance

- Reinforced by observing an event sequence **a. op. b.**
- Punished by failed anticipation.

"Link spikes":

- When goal **b!** arrives at concept **b**, and concept **op!** generates the highest goal truth among all to **b** incoming links**,** deduce and anticipate **b.** and execute **op** as a side effect, else pass subgoal **a!** to concept **a** using Deduction.

Inference Rules

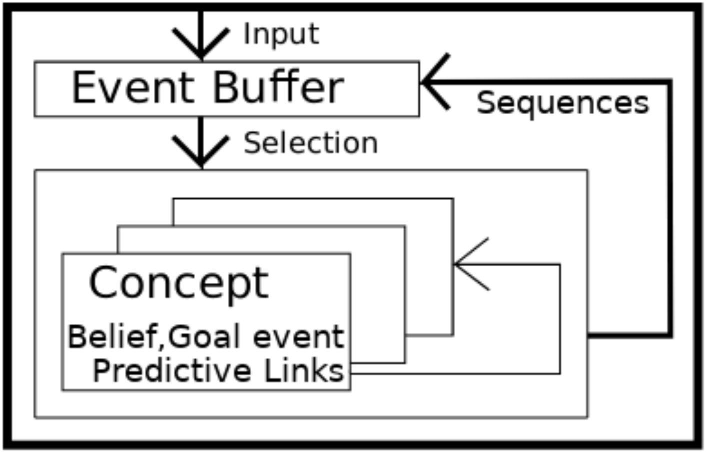
Revision and Choice:
{Event a, Event a} |- Event a
{Implication A, Implication A} |- Implication A

Intersection:
{Event a., Event b.} |- Event (&/,a,b)

Induction:
{Event a., Event b.} |- Implication <a =/> b>

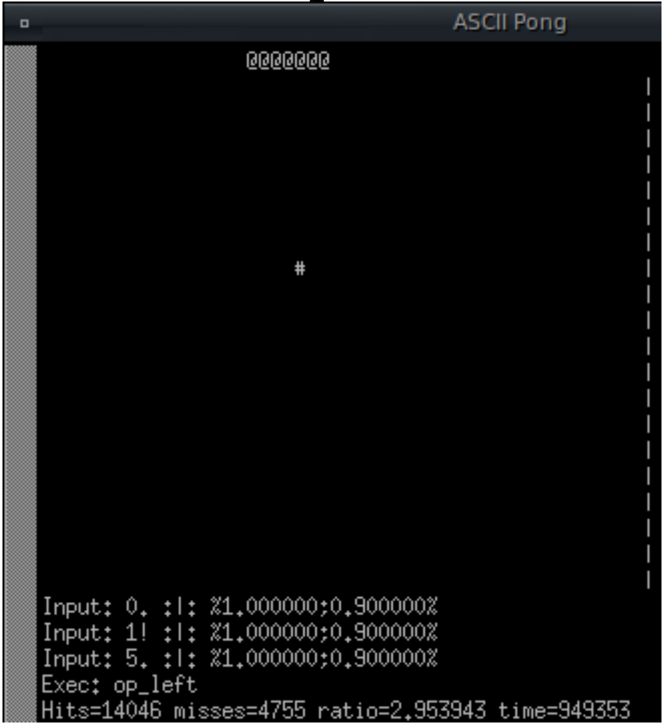
Deduction:
{Event a., Implication <a =/> b>} |- Event b.
{Event b!, Implication <a =/> b>} |- Event a!
{Event (&/,a,op())!, Event a.} |- Event op()!

Big Picture



Input
Event Buffer
Sequences
Selection
Concept
Belief,Goal event
Predictive Links

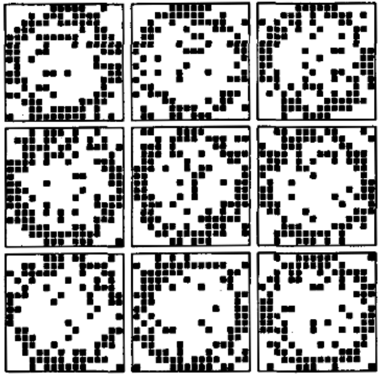
Stable C implementation

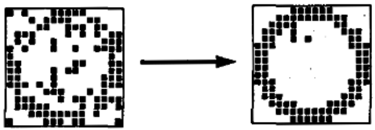


Future investigations

- Find better ways to deal with conceptual novelty than comparing event SDR's to existing concept SDR's.
- Flexible NARS-style Inheritance between concepts seems necessary to explain higher-level cognition, rather than relying on implicit/structural Inheritance given by the SDR encodings.
- Conceptual Interpolation changes existing SDR's, similar to Sparse Distributed Memory to form more representative SDR's from event matches, investigate.
- How can variable/placeholder bits be implemented efficiently, so that the system can form abstract knowledge similar to NARS?
- Hierarchically arranged receptive fields: local inhibition in each layer can automatically lead to sparse coding of input stimuli. Investigate, and in general, extend ANSNA with sensory channels.

Conceptual interpolation





Formation of a “noise-less prototype” by keeping a counter, as shown in Kanerva, P. (1988). Sparse distributed memory. MIT press

Variables in OpenNARS

- To support identity mapping, that for instance demands, associating the common property between the right and left sample as a reason for the happening satisfaction, OpenNARS introduces dependent variables:

<{right} --> [A1]>. :|:
<{left} --> [A1]>. :|:
<{SELF} --> [satisfied]>. :|:
|- (intersection then deduction)
<(&/,<{left} --> [#1]>,<{right} --> [#1]>) =/>
<{SELF} --> [satisfied]>>.

- How to realize in ANSNA?

Variable bits in ANSNA

```
left =     0101000
right =    0010010
A1 =       1000100
rightA1 =  1010110
leftA1 =   1101100
```

- Observed sequence: **leftA1, rightA1, satisfied,** hypo.:

(&/,1010110,1101100) =/> satisfied

- specific version by or-ing the bits:

(&/,rightA1,leftA1) =/> satisfied
1111110 =/> satisfied

- general version:

(&/,right(#1),left(#1)) =/> satisfied
xpspxsx =/> satisfied

Variable bits in ANSNA

(&/,right(#1),left(#1)) =/> satisfied
xpspxsx =/> satisfied

- For a new **c=(&/,a,b)** to match the precondition of this implication:

- **p** positions need to be 1 in **b**,
- **s** positions needs to be 1 in **a**,
- in **c, x** can be either 0 in **a** and **b**, or 1 in **a** and **b** both.

- So all the **x** positons together act as the variable SDR part in the SDR's **a** and **b**, which has to be the same in both **a** and **b** to match the precondition.

Thank you!

ANSNA

SENSORIMOTOR SYSTEM

Website:
https://github.com/patham9/ANSNA