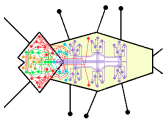
May We Have Your Attention: Analysis of a Selective Attention Task

Eldan Goldenberg, Jacob Garcowski, Randall Beer

CASE
CASE WESTERN RESERVE UNIVERSITY

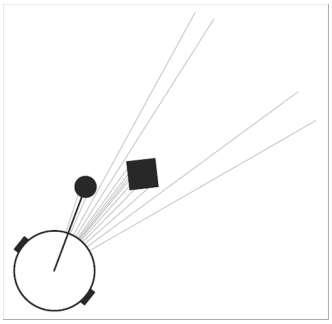


Overview

- Minimally Cognitive Agents
- Selective attention task
- Task analysis
- Controller structure and evolution
- Agent performance
- Shaping evolution

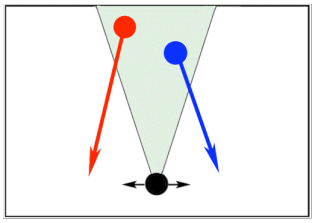
Minimally Cognitive Agents

- Focus on the simplest behaviour that raises issues of genuine cognitive interest



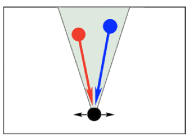
The selective attention task

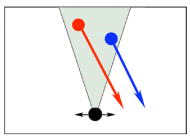
- Agent has one degree of freedom and must catch two falling objects
- The objects' speeds and trajectories are random but constrained to make both catchable

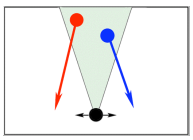


Task breakdown

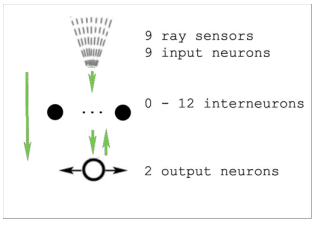
- Different parameters make different demands of the agent, including a combination of:
  - Object selection
  - Object permanence
  - Short-term memory







Controller structure

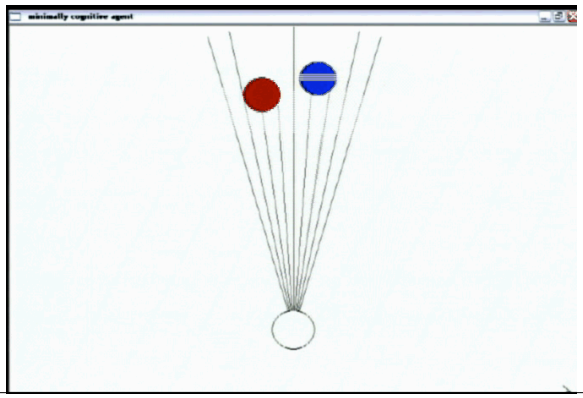


$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji}\, \sigma(g_j(y_j + \theta_j)) + I_i \quad i = 1, \ldots, N$$

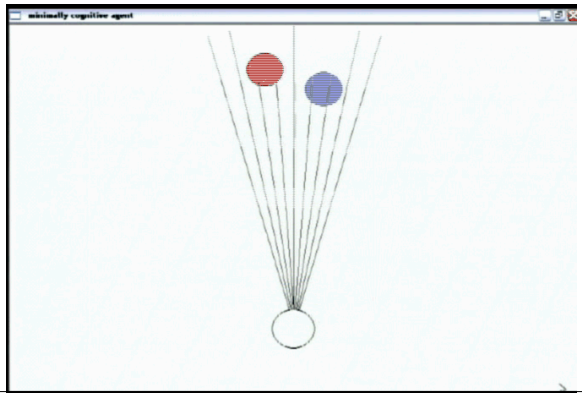
Evolution of catching

- All neural network parameters evolved
- Constrained to bilateral symmetry
- Limited trial set
- New trials added when performance is good on existing set
- Trials co-evolve with the agent to challenge it

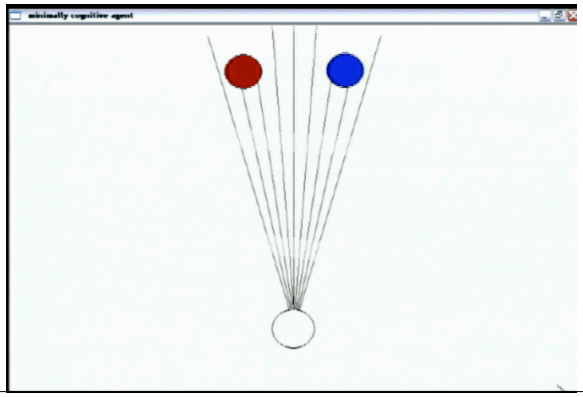
Reactive behaviour



A decision problem



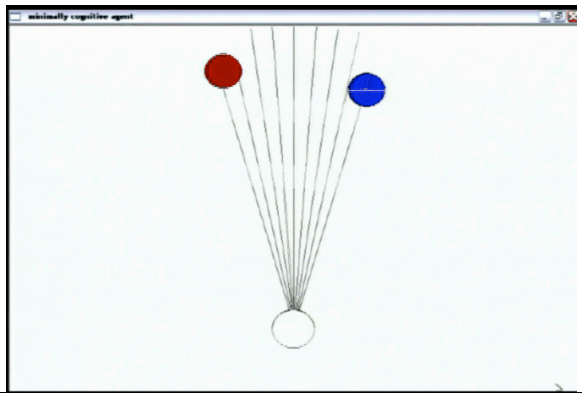
A problem requiring memory



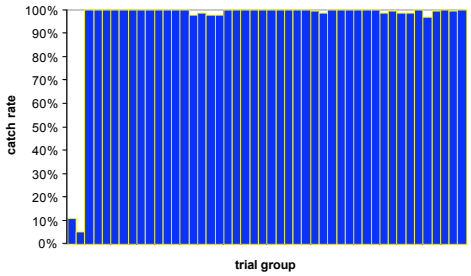
The most robust generaliser

- Overall catch rate 99%
- Striking specific deficit: catch rate of 20% in a subset of trials
- Inferred heuristic: follow the closer object when a decision must be made

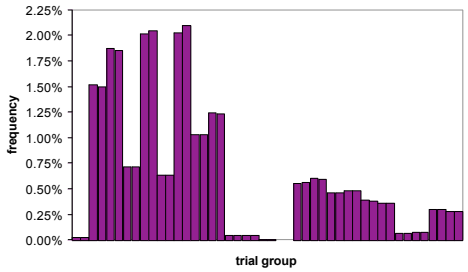
Weakness of the best agent



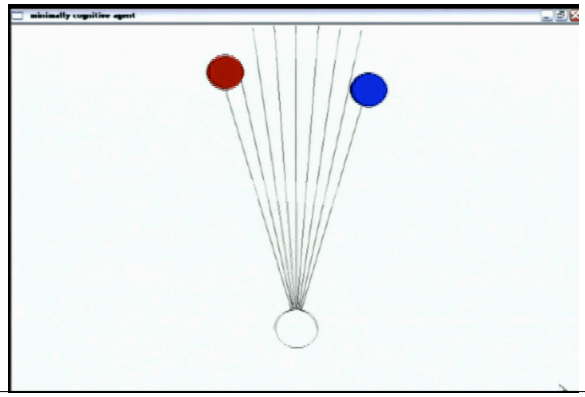
A very specific deficit



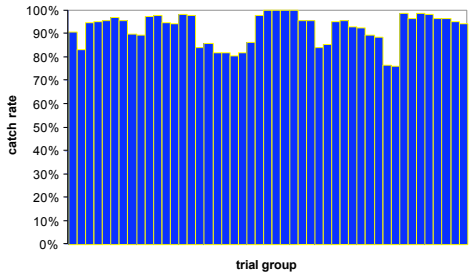
Distribution of trials



These trials are tractable



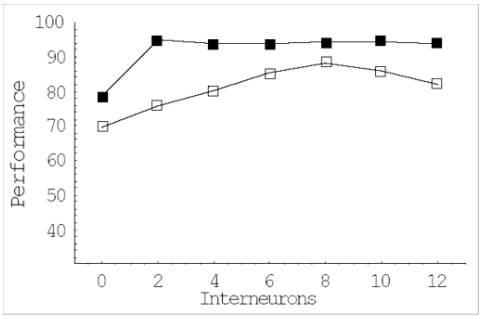
A general deficit



Identifying 'cheating' agents

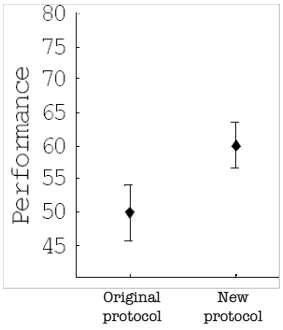
- On any given trial type it is possible to perform well without doing anything that could be described as cognitive
- But robust behaviour requires flexibility

Network size vs performance



Shaping evolution

- Different trial types require different strategies
- Not all trial types are presented in evolution
- Forcing a particular type to be presented improves performance on it



Conclusions

- Even such an apparently simple problem has a rich substructure
- Understanding the substructure is essential to interpret agents' performance
- This task produces a wide range of strategies even with a simple control architecture