CABot1: a Videogame Agent Implemented in fLIF Neurons

Christian Huyck
Middlesex University of London, UK
c.huyck@mdx.ac.uk

Abstract – *CABot1 is an agent in a simple videogame that assists a user in the game. Like the user, it views the game; it takes commands from the user, the commands are used to set goals, and the system interleaves all of these processes. Crucially, CABot1 is implemented entirely in simulated fatiguing Leaky Integrate and Fire neurons. The long term goal of this line of research is to develop a system that can solve the Turing test. The author believes that the best approach to building such a system is to mimic humans at a neural and psychological level. CABot1 makes use of the Cell Assembly hypothesis as the neural implementation of symbols, and as such, CABot1 processes symbols. Developing increasingly complex systems, like CABot1, that are grounded in an environment should lead to a system that is capable of grounding its symbols, learning new symbols, rules and relationships. If such a system is allowed to learn enough, it will be able to pass the Turing test. CABot1 does not pass the Turing test, but it does demonstrate how vision, language, and planning can be implemented in a neural system and integrated into a useful system.*

Keywords: Fatiguing Leaky Integrate and Fire Neurons, Videogame Agent, Turing Test, Neural Agent

1 Introduction

CABot1 is a prototype games agent for an EPSRC funded project. It is an agent in the game and assists the user in the game. To do this it process visual information, and the user's natural language commands; it has a goaling system and a system for controlling activity within the system. It is implemented entirely in fatiguing Leaky Integrate and Fire (fLIF) neurons.

Though CABot1 does not perform perfectly, it is an effective assistant. While a symbolic agent might be developed more easily, the use of neurons provides a hook for grounding symbols and making use of these grounded symbols. Grounding is relatively simplistic in CABot1, but it has made a step toward addressing the symbol grounding problem.

CABot1 is an early prototype for the project, and two more prototypes are planned within the project. It is hoped that this work will continue beyond the current project. The long-term goal is to address the Turing [Turing, 1950] task and develop a true Artificial Intelligence (AI). I feel that the best path toward developing a real AI is to follow the human model closely but not slavishly. It is important to develop systems that do interesting things, and adhere to known neural and psychological processes.

2 Background

AI is an ill defined concept. There is currently no machine that is intelligent like a human, and none is on the horizon. This said, AI has had incredible advances and is widely used industrially. Still, no system seems likely to pass the Turing test [Turing, 1950] in the foreseeable future. The advantages of a system that could pass the test are obvious including, for example, improved user interface, improved data mining, better videogames, intelligent assistants, intelligent monitoring, and excellent tutorial systems to name a few.

AI has had many successes over the years, and AI technologies are increasingly relevant in industry. Expert systems have been used for decades to solve specialised problems. Until the 80s, AI generally referred to symbolic systems. It was hoped that symbolic systems would lead to a system that could pass the test. Unfortunately, these systems required knowledge to be placed into them. Systems like Cyc [Lenat, 1995], had an incredible amount of time put into developing their knowledge bases, and, though useful, showed that the symbolic approach alone was unlikely to easily lead to a system that can pass the test.

Machine learning systems currently provide a host of information for industry and the public at large. Machine learning could provide the solution to the problem. The system could be programmed with some basic facts, and could then learn other things. Current machine learning systems are, however, limited to either trivial problems, very simple domains, or to using highly pre-processed or simplified data. What then would be a good way to build a system that could pass the Turing test?

One promising mechanism might be to use a cognitive model like ACT [Anderson and Lebiere, 1998]. These models are largely symbolic, but do account for a wide range of human psychological data. While ACT and other symbolic cognitive models can learn, they are restricted to combining existing symbols in novel ways. What is needed is a way to ground symbols [Harnad, 1990].

One mechanism for grounding symbols is to use simulated neurons. This approach has been used, and, though still in its infancy, is promising because it can take advantage to known results about human (and other animal) neural learning. The Cell Assembly (CA) hypothesis [Hebb, 1949] provides a useful bridge between neurons and symbols. A CA is a set of neurons that has high mutual synaptic strength that can provide a reverberating circuit; it is the neural basis of symbols. Extensive neuropsychological evidence supports this hypothesis (e.g. [Abeles et al., 1993]).

There has been a vast amount of research in neurophysiology and neuropsychology. All of this provides evidence on how to build a system. However, these studies usually relate to a small number of neurons. Understanding of emergent properties of neurons is at an early stage. One way to improve this understanding would be to build a neuro-cognitive architecture [Huyck, 2001]. Interestingly, this architecture would also provide a platform for developing a system that could pass the Turing test. After all, the Turing test is just a human trying to decide if a system is human or not.

CABot1 is not a neuro-cognitive architecture. It is a sophisticated neural system that is hoped to be a stepping stone toward such an architecture. The next section provides some reasons for the development of CABot1.

3 Reasons for CABot1

The first question is: why a videogame? In the long run, it is important that the agent is embodied in an environment. This enables the agent to interact with the environment and learn from the interaction. The real world is a rich environment and thus a robot would be an excellent agent. However, a robot requires hardware and a range of sensors and actuators. A videogame still provides sensory input, and the agent can act, but compared to a robot it is quite simple to make the action. Moreover, the videogame can be more easily controlled than the real environment.

The second question is: why neurons? The prior section provided some answers to this question. People think using neurons. Neurons can be used to learn relations, to learn new rules [Huyck and Belavkin, 2006], and for symbol grounding. Finally, neural systems are Turing complete so anything that can be programmed can be programmed in a neural system. In particular, fLIF neural systems are Turing complete; a quick proof is that any FSA can be programmed with fLIF neurons [Fan and Huyck, tted]; a stack can be implemented with fLIF neurons [Huyck and Fan, 2007]; any Turing machine can be implemented with a FSA and two stacks. This is a novel result but is not particularly surprising as others have proved that other types of connectionist systems are Turing complete.

The third question is : why fLIF neurons? fLIF neurons are a relatively accurate model of biological neurons. The integrate and fire (IF) model [McCulloch and Pitts, 1943] is a long standing biological model. Work on spin glass models of IF neurons [Hopfield, 1982] has strong mathematical foundations. These models and their extension to include leak of activation when a neuron does not fire (LIF) [Maass and Bishop, 2001] are widely used in modelling neuropsychological phenomena (e.g. [Amit, 1989]), but the mathematical foundations require that neurons are well connected and the connections are bi-directional. Connections in the brain are uni-directional and neurons are not well connected. Still LIF models using this topology are widely used in modelling neuropsychological phenomena.

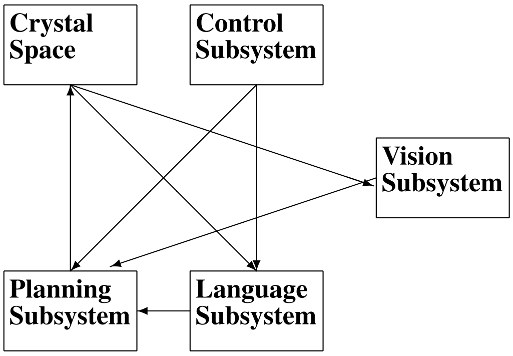

Figure 1: Interaction Between CABot1 subsystems (control, vision, planning and language) and Crystal Space.

The introduction of fatigue into the model is relatively uncommon [Huyck, 2007]. As a neurons fires, it tires and becomes more difficult to fire in the future. This fatigue extends the LIF model to the fLIF model. One benefit of fatigue is that it enables the network to more easily transition from one state to another. With a simple model designed for categorising one input, only one state is needed, but with more complex processing, the system needs to move to new states. CABot1 is an example of a neural system that does complex processing.

4 CABot1

CABot1 is a neural system that views the environment, processes language, and performs actions. The author is aware of only one other neural system that does this [Knoblauch et al., 2004]. Unlike this system, CABot1 processes a context free language, and incorporates a planning system.

The overall architecture is represented in figure 1. The agent operates in a games environment (see section 4.1) as an assistant to a user. The user issues natural language text commands that are sent to the language subsystem, and the view of the environment from the agents perspective is sent to the Vision subsystem (see section 4.2). Crystal Space gets the commands issued from the planning subsystem (see section 4.3). CABot1 consists of the four subsystems, and the control subsystem manages planning and language processing. The vision and language subsystems send information to planning. Each subsystem consists of subnets of fLIF neurons and all are explained more thoroughly below.

4.1 Crystal Space

CABot1 is an agent in a video game. The game is implemented in Crystal Space [CrystalSpace, 2008], a freely available games engine. One of the tutorials provided in the Crystal Space download was modified to be the game. The game is extremely simple and is based within a single room. There is a user, the CABot1 agent, and one item, either a pyramid or a stalactite. The user and agent each have a separate view of the room, and both can turn left or right or move forward or backward. The user can also send commands to the user via text. The agent's camera views are saved to .jpg files, the user commands are written to a file, and these are the input to CABot1. The output of CABot1 is a series of one of the four commands, left, right, forward, backward, or one of two error codes. The Crystal Space game reads these commands and the agent in the game moves.

4.2 Vision & Parsing Subsystems

The two largest subsystems of CABot1 are the vision subsystem and the parsing subsystem. As both vision and parsing subsystems have been described elsewhere, so the descriptions here are brief.

An earlier version of the vision subsystem [Huyck et al., 2006] recognised wire frame objects. This system is relatively simple consisting of a simulated retina of on-off and off-on detectors, a primary visual cortex of line and angle detectors, and a specific secondary visual cortex to recognise two shapes. The addition of edge detectors to the simulated primary visual cortex allowed the system to recognise solid pyramids and stalactites. To some degree, the simulated retina and primary visual cortex are biologically realistic. On the other hand the secondary visual cortex is a rough way to get the system to work. The active visual item resides in this area persisting in both position variant and invariant CAs.

The parsing system has been described elsewhere [Huyck and Fan, 2007]. It uses variable binding via short-term potentiation to implement a stack and verb frames. The semantic result of the parsed sentence is stored in the verb frame.

4.3 Planning & Goaling Subsystem

The planning and goaling subsystem was inspired by the Spreading Activation Network [Maes, 1989]. The fLIF implementation is based around three subnets, the fact subnet, the module subnet and the action subnet.

The fact subnet contains the facts and current goals of the agent with each fact represented by a CA. If the fact or goal is current, the CA is active with a large percentage of neurons firing. Facts can be set from the visual subsystem or from the planning subsystem itself. For CABot1, the relevant part of visual input is stored in the secondary visual cortex (see section 4.2). As symbolic facts this can be summarized as object, object size, and location. So if there is a small pyramid in the left part of the agent's visual field, the secondary visual cortex ignites the left fact, pyramid fact, and small fact CAs. These facts change as the environment changes.

Goals are also CAs in the fact subnet and are set either from the parsing subsystem or from the planning subsystem. When parsing is completed, the top stack element is activated. This leads to the activation of the semantics of the user's command in the form of a verb frame. Connections from this frame to the goals cause the appropriate goal to be activated. So if the user commanded "Go left." the $left$ goal would be set via activation from the parsing's semantic representation of the sentence.

The second subnet is the modules subnet with each module having its own CA. A module is a rule which if activated causes an action to occur, and may change the facts and goals. So, there are connections from the fact subnet to the module subnet. If the appropriate facts are active, a module will ignite. The module may have connections back to the fact subnet to change the internal representation of the world. For instance, if the user enters the command "Go right", the $go-right$ goal is set by the parsing subsystem. This requires the system to turn right and then move forward. This is accomplished by two module CAs, the first turns right, turns the initial goal off, and turns the new goal - move forward - on. The second moves forward and turns the goal off. Modules do not actually execute commands, they merely send activation to the third subnet.

The third subnet is the action subnet, consisting of six CAs, one for each of the four primitive moves, and one for both of the possible errors. If one of these CAs becomes active, a command is emitted to a file that is read by Crystal Space. The CA turns itself off so other commands can be run.

It should be noted that the plans are context sensitive and in some cases are terminated by environmental stimuli. For instance, the command "Turn toward the pyramid." is context sensitive. If the pyramid is on the left, the agent turns left, if on the right it turns right, and if in the centre, CABot1 issues an error "already facing centre".

The command "Go to the stalactite." can only be filled when the agent gets to the stalactite. Thus a series of commands needs to be emitted. The plan tries to centre the stalactite in the visual field while moving forward. It only stops when the stalactite becomes very large in the visual field.

4.4 Control Subsystem

The control subsystem manages the interactions between the other three subsystems. It is a single subnet consisting of five CAs. The CAs are arranged so that a loop is executed. The loop is:

1. Get Command

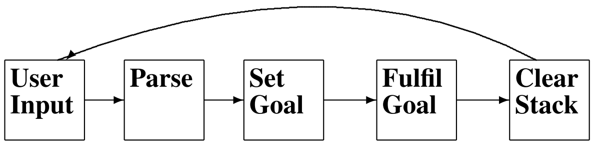

Figure 2: Flow of Control of Control Subsystem.

2. Parse Command
3. Set Goal
4. Fulfil Goal
5. Erase Parse Stack

This is represented by figure 2. The overall behaviour of the agent is simply to wait for commands, parse them, and then execute them.

Initially the system waits for a command from the user. When one is received parsing begins. When parsing completes (the $S \rightarrow VPperiod$ rule is applied) the goal is set by the control CA activating the first parse stack item. When a goal is set, an appropriate fact subnet CA is active, and the control subsystem moves on. When no goal remains, the control net moves on to erasing the parse stack, to allow another sentence to be parsed. When the erasing is completed, the loop starts again.

During this process inhibition is used to suppress other activities. So during parsing, the planning nets are suppressed so no actions occur, and during goal fulfilment, parsing is suppressed.

This control does not directly affect vision. Vision is constantly active, but its effect on facts only registers during the set and fulfil goal phases.

4.5 Evaluation

A range of evaluations have been performed on CABot1. Several evaluations have been done on the subsystems, but here the evaluations of the performance of the entire agent are briefly described.

Firstly, it must be noted that there is some degree of randomness in the creation of a particular agent. That means that some nets will perform differently than other nets. Fortunately, nets can be stored and reread so that stable agents can be evaluated.

CABot1 processes ten commands that can be grouped into four basic categories. Performance within category is largely consistent but performance between categories varies. The simplest command is to do a direct action. There are four of these: “Turn left.”, “Turn right.”, “Move forward.”, and “Move backward.”. The best nets get these commands correct over 90% and the average nets get around 80% of these correct. So if the command is “Turn left.”, the agent emits the *left* command to Crystal Space. The failures are largely due to the planning subsystem, but some problems arise in parsing.

There are two compound commands which require two actions. These are “Go left.” and “Go right.” and the correct output is the command “left” (or “right”) followed by a “forward” command. The best nets perform correctly 85% of the time and the average nets do the right thing 75% of the time. The extra failures are due to problems in the planning subsystem. For both of these categories, vision failures do not lead to problems as they are not involved in the process. Throughout, control problems rarely arise.

The third category of commands are context sensitive commands, and there are two: “Turn toward the pyramid.” and “Turn toward the stalactite.” The correct output is the command “left” or the command “right” depending on where the object is in the agent’s visual field. The best nets get 75% of these correct and the average performance is around 60%. Vision problems arise here, but the increased parsing complexity also leads to some problems. Note also that these tests require different contexts. The system must present the desired object to the left, right, and centre, no object, and the opposite object. It also has to output the appropriate error - “already facing centre”, or “object not present”- in the appropriate case.

The final category requires feedback and these two are “Go to the pyramid.” and “Go to the stalactite”. The correct output here is a series of actions that result in the agent being adjacent to the object. The best nets achieve the goal roughly 50% of the time, and the average net does so 35% with some nets unable to achieve the goal. One problem is over correction during centring.

What is clear from these experiments is that the system works, though it is far from perfect. It also works on a rather small set of commands and in an impoverished environment. While the parser could easily accommodate more commands (it already handles 23 commands correctly up to 99% of the time) and control works almost all the time, vision and planning are more constrained. The secondary visual cortex is set to work for two shapes only; other objects could be programmed, but that might lead to interference. Planning can be improved and expanded relatively easily requiring specific modules to be designed and implemented.

The results are satisfactory to show that the CABot approach works in principle. While the particular system could be improved for performance, a more general approach based around learning will be more effective in the long run.

5 Future Work & Conclusion

CABot1 was the first prototype of three planned prototypes on the current project. It has been successful at integrating language processing, visual processing, and planning into a single agent that is implemented in simulated neurons.

CABot2 is currently being implemented with a target completion date of July 2008. The focus of this agent is to use CABot1 as a foundation for exploring learning. All learning is done via Hebbian synaptic weight modification. Two learning tasks have already been completed. The first is a semi-supervised labelling task where the visual concepts pyramid and stalactite have the appropriate labels attached via co-presentation. That is the word pyramid is presented along with a visual pyramid, and this association is learned. A second task is to learn the correct action for a given goal. This is done via feedback from the environment and this may provide the basis for learning new modules.

Two further tasks are being implemented. The first is prepositional phrase attachment ambiguity resolution. This is implemented with semi-supervised learning with correct attachments being supplied. This use hierarchical learning to allow generalisation. This hierarchy requires the parser being rewritten. As the CABot1 parser has timing problems (it takes a lot of simulated time to parse a sentence), a more efficient parser without a stack is being implemented. The second task for learning is to learn new visual objects.

CABot3 is in the planning stages now. The target date for completion is May 2009 with the grant finishing at the end of July 2009. It is hoped that this agent will be based in a more sophisticated game environment so that it can learn from the environment. It is also hoped that it will be more interesting to play.

A follow on grant is envisioned starting in 2009 and finishing in 2012. This will probably have multiple prototypes called CABot4 etc. The plan is that these agents will be conversational agents so that they can both understand and produce language. These should have more sophisticated cognitive mapping ability to build up sophisticated domain representations. It is also hoped that these agents will mimic more psychological data.

A similar goal is to build a neural cognitive architecture. This will be a growing body of work where more and more psychological phenomena are accounted for by the architecture and one or more implementations. Systems built on this architecture should be developed in more and more domains. As humans can learn new domains, it is hoped that these systems will be able to learn new domains. One prominent goal is passing the Turing test. This work will benefit from a community of developers using this and related architectures.

The long term objective is to develop a system that can pass the Turing test [Turing, 1950]. Eventually, these agents will move from the virtual world to the real world in the form of robots; this will require a range of motor and sensory control that can be avoided in virtual environments. However, this will give the agent access to a new range of haptic and motor semantics. Reaching the long-term objective will require a long range effort with a host of problems to be solved along the way; indeed it is highly unlikely that this effort will succeed without participation from more than one research group. However, by using human neural and psychological performance as guideposts, and developing ever more capable systems, this approach can succeed.

CABot1 can be downloaded from http://www.cwa.mdx.ac.uk/CABot/CABot.html. CABot2 should be available by the time of publication. The author encourages anyone interested in this project to contact him via email.

Acknowledgements

This work is supported by EPSRC grant EP/D059720 - Natural Language Parsing using Cell Assemblies: Computational Linguistics with Attractor Nets.

References

[Abeles et al., 1993] Abeles, M., Bergman, H., Margalit, E., and Vaadia, E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *Journal of Neurophysiology*, 70(4):1629–38.

[Amit, 1989] Amit, D. (1989). *Modelling Brain Function: The world of attractor neural networks*. Cambridge University Press.

[Anderson and Lebiere, 1998] Anderson, J. and Lebiere, C. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum.

[CrystalSpace, 2008] CrystalSpace (2008). *Crystal Space*. $http : //www.crystalspace3d.org/main/Main_Page$.

[Fan and Huyck, tted] Fan, Y. and Huyck, C. (submitted). Implementation of finite state automata using flif neurons. In *IEEE Systems, Man and Cybernetics Society*.

[Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

[Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior*. J. Wiley & Sons.

[Hopfield, 1982] Hopfield, J. (1982). Neural nets and physical systems with emergent collective computational abilities. *Proc. of the Nat. Academy of Sciences USA*, 79:2554–8.

[Huyck, 2001] Huyck, C. (2001). Cell assemblies as an intermediate level model of cognition. In Wermter, S., Austin, J., and Willshaw, D., editors, *Emerging Neural Architectures based on Neuroscience*, pages 383–397. Springer.

[Huyck, 2007] Huyck, C. (2007). Creating hierarchical categories using cell assemblies. *Connection Science*, 19:1:1–24.

[Huyck and Belavkin, 2006] Huyck, C. and Belavkin, R. (2006). Counting with neurons: Rule application with nets of fatiguing leaky integrate and fire neurons. In *Proceedings of the Seventh International Conference on Cognitive Modelling*, pages 142–147.

[Huyck et al., 2006] Huyck, C., Diaper, D., Belavkin, R., and Kenny, I. (2006). Vision in an agent based on fatiguing leaky integrate and fire neurons. In *Proceedings of the Fifth International Conference on Cybernetic Intelligent Systems*.

[Huyck and Fan, 2007] Huyck, C. and Fan, Y. (2007). Parsing with flif neurons. In *IEEE Systems, Man and Cybernetics Society*, pages 35–40.

[Knoblauch et al., 2004] Knoblauch, A., Markert, H., and Palm, G. (2004). An associative model of cortical langauage and action processing. In *Proceedings of the Ninth Neural Computation and Psychology Workshop*.

[Lenat, 1995] Lenat, D. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communicatiosn of the ACM*, 38:11:33–38.

[Maass and Bishop, 2001] Maass, W. and Bishop, C. (2001). *Pulsed Neural Networks*. MIT Press Cambridge MA.

[Maes, 1989] Maes, P. (1989). How to do the right thing. *Connection Science*.

[McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

[Turing, 1950] Turing, A. (1950). Computing machinery & intelligence. *Mind*, 59:236:433–60.