RatCog: A GUI maze simulation tool with plugin "rat brains"

Christopher G. Prince[1], John Talton[2], István S. N. Berkeley[3], &
Cengiz Gunay[3]

[1]University of Minnesota Duluth, [2]Louisiana Tech University, and
[3]University of Louisiana Lafayette

Corresponding author: Christopher G. Prince,
University of Minnesota Duluth
Department of Computer Science
320 Heller Hall, 10 University Drive
Duluth, MN 55812-2496 USA
(218) 726-7607
chris@cprince.com

Abstract: We have implemented RatCog, a Graphical User Interface (GUI) radial-maze simulation tool providing various computational models of rats. Rat models are loaded as runtime plugin files, and an Application Programming Interface (API) enables additional plugins to be created. One implemented plugin is a back-propagation trained connectionist model. GUI features include maze graphics and performance statistics. The GUI makes it easier to use these computional models, while the plugins make the models widely available.

Paper accepted for presentation at: SciP (Society for Computers in Psychology) 2000 Conference, to be held in New Orleans, November 16th, 2000.

Summary:

The RatCog project addresses two needs in the modeling of rat cognitive processes: the need for visualization and the need for portability. There is a need for visualization and graphical display of the simulations. While many computational models of rats (e.g., Benhamou, Bovet, & Poucet, 1995; Burgess, Recce, & O'Keefe, 1994; Guazzelli, Bota, Corbacho, & Arbib, 1998; Redish, 1999; Shapiro & Hetherington, 1993; Sharp, 1991; Trullier & Meyer, 1998; Zipser, 1985) have been aimed at issues relating to neuroscience, in order for these models to become widely available to practicing psychologists, a Graphical User Interface (GUI) needs to be provided. Just as windowing-based operating systems (e.g., MacOS™) are often easier to use than text-based operating systems (e.g., UNIX), it also becomes quickly apparent that models with text-based output are often unintelligible to everyone (except their authors)! Visualization of computational models includes neuroscience issues (e.g., hippocampal neural cell or "place cell" visualization, Maida, Yuen, & Prince, 1996; Maida & Yuen, 1999), and RatCog complements this approach by addressing visualization and graphical display of rats on a maze and their associated performance data.

There is also a need for enhanced portability of rat computational models. When a practicing psychologist reads a book on computational models of rats (e.g., Schmajuk, 1997), it would be tremendously useful for them to be able to use these models in addressing their own research questions. Unfortunately, using other researcher's models quite likely involves re-implementation, or porting program code. Even experienced computer scientists can find this task daunting (see also Maida & Yuen, 1999). RatCog addresses this issue by providing a standardized plugin architecture. A plugin architecture defines a standardized interface, the Application Programming Interface (API) for the plugins, enabling other programmers to create new plugins and hence add additional functionality to the system. New plugins are programmed by adhering to the API. "Rat brain" plugins designed for RatCog can be downloaded off the Internet, and rapidly used to address a researchers own problem of interest.

Methods

RatCog has been implemented in the C++ programming language on the BeOS operating system. BeOS is a windowing-based operating system available for Intel and PowerPC hardware, and has a "Personal Edition" that is a free download on the Internet (see http://www.be.com), and installs easily as a 500 megabyte file under Microsoft Windows™ with no disk repartitioning necessary. RatCog provides a "software agent" environment (Russell & Norvig, 1995; see also Gunay, 2000) for rat-like intelligent agents. It is designed using a client-server approach to intelligent agent simulation (e.g., see the RoboCup Soccer Server, Noda, 1995).

The software architecture of RatCog is divided into four components: the GUI, the database, the world model, and the "rat brain" plugins. The GUI displays a radial-maze (see Olton & Samuelson, 1976), and it enables user control of the simulation while also displaying performance statistics. The database component enables the user to store and retrieve information about the simulation such as experiment settings (number of rats, trial scripts, etc.), the number of maze arms (between 2 and 16), settings relevant to the plugins (e.g., learning rate, number of hidden layers), and results of experiments. The world model provides the simulation of the maze and keeps track of the food on the maze and the current position of the rat. Through the API, the "rat brain" plugin can query the world model (just like a real rat can perceive the environment), and so the API provides access both to simulated sensory input and simulated motor output via interaction with the world model. The "rat brain" plugins perform the actual work of controlling the simulated rats.

In addition to a random-rat plugin which just chooses arms on the maze at random, we have implemented a connectionist model plugin, that is trained with the back-propagation technique (Rumelhart, Hinton, & Williams, 1986). Connectionist models simulate some abstract properties of biological neural systems and moreover often have the ability to learn. Our connectionist plugin represents places (positions on the maze) as vectors of eight real numbers. The plugin provides three kinds of models, differentiated according to their type of output (all models have a place as input). The three kinds are: place-place, place-response, and place-mixed models. Response outputs are represented as length two vectors (LEFT or RIGHT turns), and a "mixed" output refers to a length 10 output vector concatenating a place and a response.

Results

We are in the process of simulating a complex maze task (Prince, 1998) by using RatCog. The task in Prince (1998) involves rats learning a pair of paths on a four-arm radial-maze, and then learning the reverse of this pair of paths. Prior simulations of this task based on place and response hypotheses (Restle, 1957) have suggested that neither place nor response hypotheses nor a combination of place and response hypothesis closely model the actual performance of rats on this task (Berkeley, Prince, & Gunay, 1999). The present plugin is aimed to replicate and extend the models of Berkeley et al. (1999). These prior models utilized a back-propagation trained connectionist model with no hidden layer. Our plugin has a setting enabling inclusion of a variable number of hidden layer units.

Discussion

This tool should assist practicing rat psychologists in modeling their tasks, and should positively impact their empirical research. The GUI should make the models easier to use, and the plugin architecture should enable rapid dissemination of various models. While other rat models in this context have had pedagogical goals (Graham, Alloway, & Krames, 1994), RatCog is targeted at the researcher. Planned future work includes extending the environment features to provide a plugin architecture. For example, RatCog currently is coded to simulate a radial-maze. Other mazes, such as the Morris water maze (Morris, 1981) would be valuable to include in the environment simulation. Other “rat brain” plugins that we have planned include a multi-food type foraging model to simulate a rat’s performance on chunking tasks (e.g., Macuda & Roberts, 1995), the Neath and Capaldi (1996) simulation, and the Burgess et al. (1994) hippocampal model.

Acknowledgements

The authors would like to thank Rich Maclin and Ben Choi for their comments on this paper. Tony Maida stimulated much of these activities, and Danny Povinelli encouraged persistence.

References

Benhamou, S., Bovet, P., & Poucet, B. (1995). A model for place navigation in mammals. Journal of Theoretical Biology, 173, 163-178.

Berkeley, I. S. N., Prince, C., & Gunay, C. (1999). RATNET: Exploring Rat Navigation With Artificial Neural Networks. Paper presented at the 1999 Southern Society for Philosophy and Psychology Meeting in Louisville, Kentucky.

Burgess, N., Recce, M., & O'Keefe, J. (1994). A model of hippocampal function. Neural Networks, 7, 1065-1081.

Graham, J., Alloway, T., & Krames, L. (1994). Sniffy, the virtual rat: Simulated operant conditioning. Behavior, Research Methods, Instruments, & Computers, 26, 134-141.

Guazzelli, A. Bota, M., Corbacho, F. J., & Arbib, M. A. (1998). Affordances, motivations, and the world graph theory. Adaptive Behavior, 6, 435-471.

Gunay, C. (2000). Autonomous agent navigation in grid world: Using java exceptions in a nonstandard way and an object oriented approach to evolution of intelligence. In Proceedings of IEA/AIE-2000, Conference held in New Orleans, LA, June 19-22.

Maida, A. S., Yuen, G. L., & Prince, C. G. (1996). Visualization of neurodynamics in a model for spatial navigation. Proceedings of the World Congress on Neural Networks (pp. 694-697). Conference held in San Diego, California, September 15-18. Mahway, NJ: Lawrence Erlbaum, Inc.

Maida, A. S. & Yuen, G. (1999). Identifying causal structure in a biologically constrained connectionist network. Unpublished manuscript. Available: ftp://basin.cacs.usl.edu/pub/uslstuff/maida/causalStructApr20.pdf

Macuda, T. & Roberts, W. A. (1995). Further evidence for hierarchical chunking in rat spatial memory. Journal of Experimental Psychology: Animal Behavior Processes, 21, 20-32.

Morris, R. G. M. (1981). Spatial localization does not require the presence of local cues. Learning and Motivation, 12, 239-260.

Neath, I. & Capaldi, E. J. (1996). A “random-walk” simulation model of multiple-pattern learning in a radial-arm maze. Animal Learning and Behavior, 24, 206-210.

Noda, I. (1995). Soccer server: A simulator for RoboCup. Paper presented at the Japanese Society for Artificial Intelligence Symposium 95: Special Session on RoboCup, Dec. 1995.

Olton, D. S. & Samuelson, R. J. (1976). Remembrance of places passed: Spatial memory in rats. Journal of Experimental Psychology: Animal Behavior Processes, 2, 97-116.

Prince, C. G. (1998). Rats’ capabilities with relations between locations in space. Unpublished doctoral dissertation, University of Southwestern Louisiana. Available: http://www.cprince.com/dissertation/

Redish, A. D. (1999). Beyond the Cognitive Map: From Place Cells to Episodic Memory. Cambridge, MA: MIT Press.

Restle, F. (1957). Discrimination of cues in mazes: A resolution of the ‘place-vs.-response’ question. Psychological Review, 64, 217-228.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533-536.

Russell, S. J. & Norvig, P. (1995). Artificial Intelligence: A Modern Approach. NJ: Prentice Hall.

Schmajuk, N. A. (1997). Animal Learning and Cognition: A Neural Network Approach. New York, NY: Cambridge University Press.

Shapiro, M. L. & Hetherington, P. H. (1993). A simple network model simulates hippocampal place fields: Parametric analyses and physiological predictions. Behavioral Neuroscience, 107, 34-50.

Sharp, P. E. (1991). Computer simulation of hippocampal place cells. Psychobiology, 19, 103-115.

Trullier, O. & Meyer, J.-A. (1998). Animat navigation using a cognitive graph. In R. Pfeifer, B. Blumberg, J.-A. Meyer, & S. W. Wilson (Eds.), From Animals to Animats 5 (pp. 213-222). Cambridge, MA: MIT Press.

Zipser, D. (1985). A computational model of hippocampal place fields. Behavioral Neuroscience, 99, 1006-1018.