

Teaching Cognitive Science with MATLAB at Stanford University

By Professor James L. McClelland, Stanford University

MANY COGNITIVE SCIENTISTS BELIEVE THAT CAPABILITIES LIKE LANGUAGE, perception, and memory are distributed over a large number of neurons operating in parallel, and that learning occurs through the adaptation of connections between these neurons. Using a Parallel Distributed Processing (PDP) framework, researchers simulate neural networks at various levels of abstraction, visualize their activities, and perform statistical and mathematical analysis to gain further insights. While the PDP framework has opened new doors for cognitive scientists and researchers, students at all levels can have a difficult time understanding the operation of such networks.

When I began using PDP networks in my teaching more than 20 years ago, it quickly became clear that for students, simply having the networks described was a poor substitute for seeing them in action. PDP simulations involve integration and optimization over large systems of nonlinear differential equations, making them difficult for students to grasp. For the instructor, the challenge is to avoid overwhelming new students with the computational complexity of the simulations while enabling advanced students to perform more sophisticated explorations.

In *PSYCH 209, The Neural Basis of Cognition: A Parallel Distributed Processing Approach*, my students use MATLAB® simulations to explore cognitive processes. MATLAB enables them to visualize the networks as they evolve over time and perform simulation and analysis in a single environment.

Moving “the PDP Class” to MATLAB

At Carnegie Mellon in the 1980s, my colleagues and I developed a set of text-based simulation tools that I used to teach what became known as “the PDP class.” While these tools helped me demonstrate how PDP networks worked, they had serious limitations. They only enabled students to explore concepts in the context of very small networks. The tools could not be used to investigate more advanced processes, such as language processing and reading, because they did not fully support some of the newer types of networks. Another drawback was that students found it difficult to visualize their simulation results because these were typically just rows of numeric values.

In addition to enabling the students to examine more complex processes, I wanted the PDP framework to make it easier for

them to pursue their own ideas and analyze the results of their network simulations. That is why, when I began teaching the class at Stanford, I reimplemented the PDP software using MATLAB.

MATLAB is well-suited to this application. The concept of parallel distributed processing is closely related to vector matrix algebra. Networks include a pattern of activation, which is a vector, and a matrix of connection weights, which is a matrix. MATLAB is built around these kinds of structures.

There were other advantages to reimplementing the framework in MATLAB. First, many researchers are interested in computational approaches to cognitive neuroscience, including functional brain imaging researchers and my colleagues at the Center for the Neural Basis of Cognition use MATLAB. I want to encourage more researchers to use PDP models, and selecting

Products Used

- MATLAB®
- MATLAB Compiler™

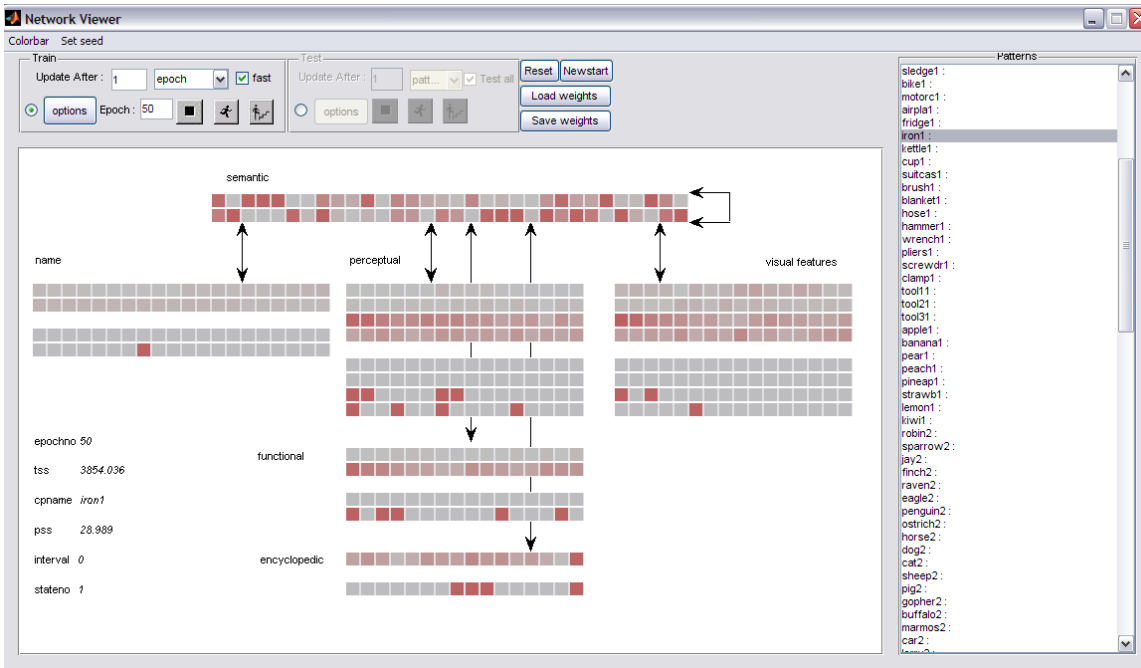


Figure 1. A PDP network, implemented in MATLAB, for simulating semantic processing.

a widely used development environment furthers that goal. Second, the simulations automatically create a MATLAB data object that can be further analyzed and displayed using the full set of visualization and analysis tools available within the MATLAB environment. Finally, as my students complete assignments and class projects, they gain experience with MATLAB that will give them an edge should they continue working in the field.

Using MATLAB in the Classroom

While some students begin Psych 209 already familiar with MATLAB, many have little or no prior experience. We provide these students with a condensed tutorial on MATLAB concepts, including expressions and vector representations. Most students quickly become comfortable with the MATLAB environment. By the time they have completed one or two homework assignments, they are able to use MATLAB to run simulations and analyze the results.

I sometimes use MATLAB to conduct a live demonstration of a simulation process. For example, we might use MATLAB visualizations to understand how a Boltzmann machine (a neural network that can find the best solution to a problem from many good ones) evolves over time. I have the students explain the MATLAB plots and graphs to make sure they really comprehend the concepts.

MATLAB plays an even larger role in the homework, particularly as we move beyond the required assignments. New students can quickly visualize preconfigured networks, while more advanced students can dig into the results using sophisticated, built-in data analysis capabilities.

In one assignment, students explore how different kinds of knowledge are integrated into a unified semantic representation. For example, the taste and smell of an apple; the ability to spell, pronounce, and recognize the word “apple”; and the fact that apples are often red, are all part of the semantic

representation of “APPLE.” Our PDP model of semantic cognition allows us to understand how semantic representations are established in childhood and how they break

About the Author

James L. McClelland received his Ph.D. in Cognitive Psychology from the University of Pennsylvania in 1975. He served on the faculty of the University of California, San Diego, before moving to Carnegie Mellon in 1984, where he became a University Professor and held the Walter Van Dyke Bingham Chair in Psychology and Cognitive Neuroscience. He was a founding Co-Director of the Center for the Neural Basis of Cognition, a joint project of Carnegie Mellon and the University of Pittsburgh. In 2006 he moved to Stanford University, where he is now Professor of Psychology and founding Director of the Center for Mind, Brain and Computation.

down in a debilitating neuropsychological disorder known as semantic dementia.

New students can start exploring a model of a semantic representation right away by invoking a MATLAB script that creates the network and displays it in the Network Viewer (Figure 1). Students use the Network Viewer to visualize the structure of the PDP network and monitor its activity. (The color of the squares changes over time as the state of the network evolves.) Students can customize this visualization using a graphical editor to display different aspects of system behavior.

More advanced students explore the effects of changing the network structure. For example, to create a new network they can use a simple point-and-click interface that creates a MATLAB script (Figure 2).

They can then examine and edit this script and call it repeatedly to run simulations for different configurations and parameter values. The script creates a data structure to represent the whole network, and the students can visualize the structure (Figure 3), examine the matrices, and write MATLAB expressions at the command prompt to evaluate different scenarios.

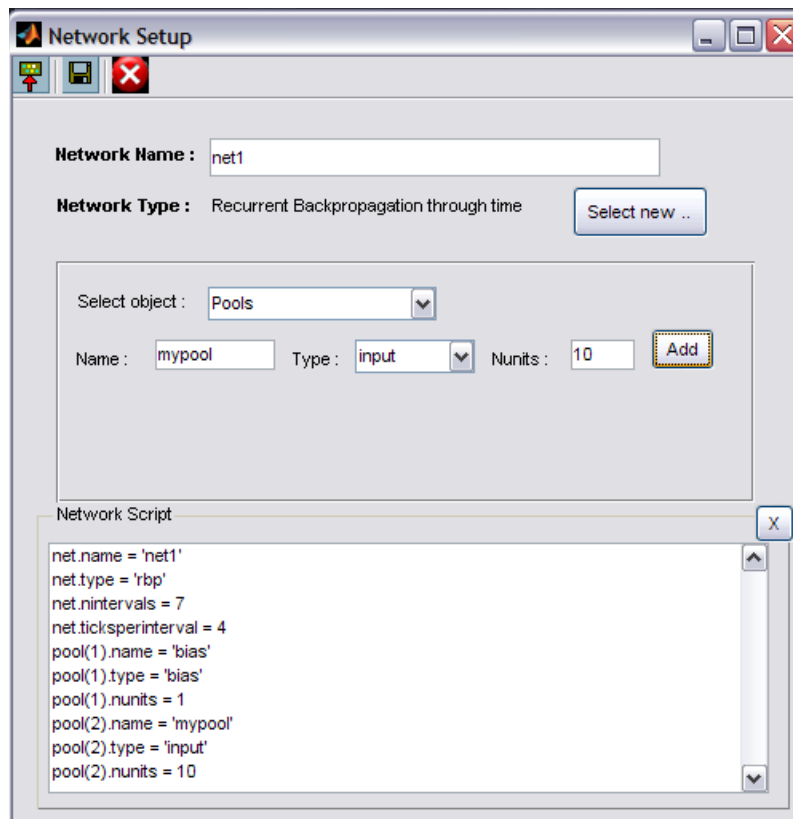


Figure 2. The graphical user interface for setting up a new network. As the student configures the network, the corresponding MATLAB script is automatically generated.

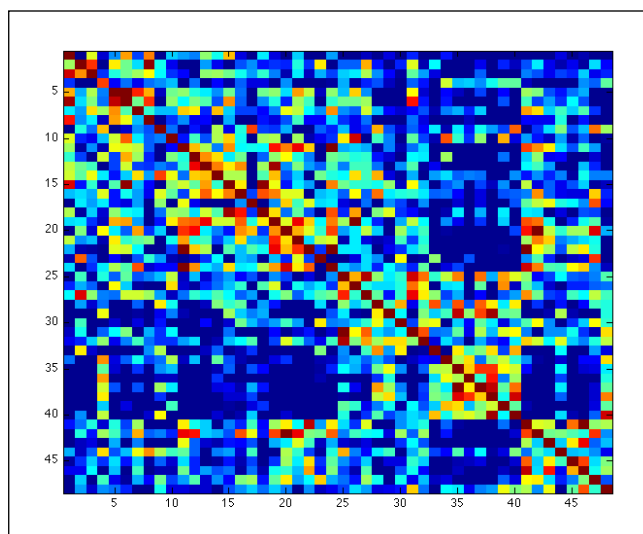
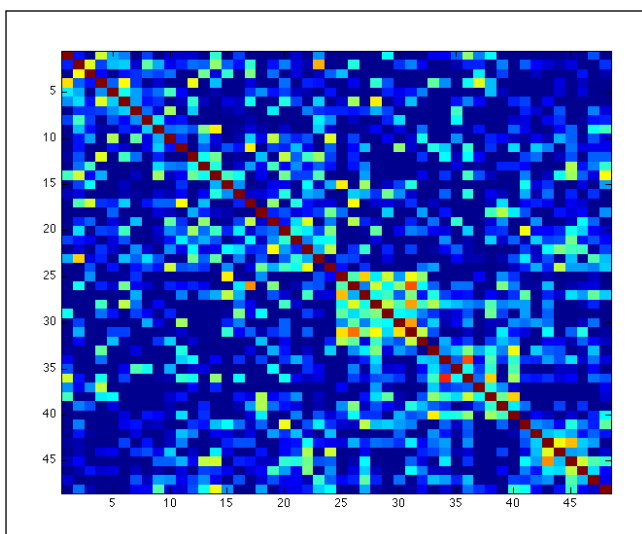


Figure 3. MATLAB visualizations representing learned patterns of activation from a trained semantic network. Left: Correlation matrix immediately after learning and before the network receives a simulated brain lesion. Right: Correlation matrix after the simulated lesion.

A Platform for Independent Research

I encourage my students to view my course not merely as an introduction to PDP but as an opportunity to engage in this way of researching cognitive processes. At the end of the first quarter they complete a mini-project on a topic of their choosing—sometimes incorporating MATLAB scripts created by researchers at other institutions.

Students who continue into the second quarter complete more extensive projects in which they use MATLAB to develop their own applications. In the past, some of these projects have culminated in articles in high-profile scientific publications. After completing the first- and second-quarter courses, some students continue to use the framework in their undergraduate and graduate research projects.

Other tools are available for doing this kind of research, including Light and Efficient Network Simulator (LENS) and Emergent. While these C- and C++-based environments were created specifically for neural network modeling and provide their own scripting languages, they do not offer the vast library of functions available in MATLAB. With MATLAB, all the tools the students need are available in one interactive environment. Furthermore, after a little experimentation with the use of embedded C code for the inner loops of neural network learning algorithms, we achieved MATLAB simulation performance comparable to the very fast LENS implementation.

I believe that the PDP framework that we created with MATLAB removes many barriers to studying cognitive neuroscience with parallel distributed processing. It is an invaluable tool not only for teaching but also for research—including my own. Using MATLAB Compiler™ we

have created standalone versions of some of the framework programs so that they can be used by researchers without access to MATLAB. We continue to improve the programs, and my sense is that there are many more advantages to using MATLAB that we have yet to fully exploit. ■

For More Information

- Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises
www.stanford.edu/group/pdplab/pdphandbook/
- MATLAB and Simulink Based Books: Biosciences
<http://www.mathworks.com/books/neuroscience>

Download

- The MATLAB-based PDPTOOL software is available for download at
www.stanford.edu/group/pdplab/pdptool/pdptool.zip

Resources

VISIT

www.mathworks.com/academia

TECHNICAL SUPPORT

www.mathworks.com/support

ONLINE USER COMMUNITY

www.mathworks.com/matlabcentral

DEMOS

www.mathworks.com/demos

TRAINING SERVICES

www.mathworks.com/training

THIRD-PARTY PRODUCTS AND SERVICES

www.mathworks.com/connections

Worldwide CONTACTS

www.mathworks.com/contact

E-MAIL

info@mathworks.com

© 2009 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

91769v00 09/09