CMPnet The Technology Network

TECHWEB

Search news:

Search
Advanced Search

News
Technology
Stocks & Finance
Internet

Resources
Web Development
Encyclopedia
Company Profiles
Events/Shows
Download Center
Product Reviews
Buy Software
Free-Product Info

E-mail Newsletters
Send Feedback

Click Advertisement
Top Headlines

• Poll: IT Outsourcing Shows No Signs Of Slowing
• Byter Up: Ballparks Go High-Tech
• AT&T Wins Breakthrough IP Contract With China
» more headlines

Free E-mail
Login - Sign Up Now

Click
Get

TechSearch

Search Home Advanced Search Search Help Web Search

EETIMES

September 21, 1998, Issue: 1026
Section: Technology

Backgammon, anyone? Neural learning theory tested

Sunny Bains

Madison, Wis. - A computer took on a human grand master and won 99 of 100 games at the American Association of Artificial Intelligence meeting held here recently. Though the system used was developed at IBM Corp.'s T.J. Watson Research Center, this time the game was not chess, but backgammon. Also, unlike Deep Blue, the machine playing was not running a conventional computer program, but a neural network using "temporal-difference" reinforcement learning. Researchers say that the success of "TD-Gammon" has been so striking that it has led to renewed interest in systems that use this type of learning scheme.

TD-Gammon's inventor is Dr. Gerald Tesauro, a researcher at the Watson Research Center in Yorktown Heights, N.Y. Tesauro has spent years developing the machine to the point where it can play this well.

According to Tesauro, "The main difference between TD-Gammon and conventional programs is that TD-Gammon programs itself, that is, it develops its own strategy and positional understanding by playing lots of games against itself." On the other hand, he said, "conventional programming is an enormously labor-intensive process whereby the programmer tries to gather knowledge from human experts, and encode that knowledge in a handcrafted evaluation function."

The advantages of doing things this way, he said, are threefold. First, "The computer does the work of coming up with a strategy, rather than the human programmer." Second, said Tesauro, "TD-Gammon's evaluation function is much better in quality than the hand-programmed functions: I would state categorically that no programmer could come close to doing by hand what TD-Gammon did by teaching itself."

Tesauro's colleague Terrence Sejnowski, a professor of biology at the University of California, San Diego, backs up that claim. "When Gerry Tesauro and I started developing an earlier backgammon program that relied on human evaluation of moves, it appeared that the program would never become more clever than the human experts that were providing the evaluation.TD-Gammon has broken this barrier," said Sejnowski, "by showing that self-play can allow the program to become better at the game than its creator."

This brings up the third advantage Tesauro claims for his system. "With conventional programming, the best that one can hope for is to equal the ability of the human expert, whereas by teaching itself, TD-Gammon can innovate, and can discover new knowledge and new strategies that actually surpass human experts' current knowledge and understanding of the game," he said.

Pushing power envelope

Tesauro admits that TD-Gammon requires a lot of computer power, both to train, and to use during play. However, he said, "with computer power doubling every year, the notion of what constitutes 'a lot' is rapidly changing and, in fact, today's high-end Pentium machines can train a TD-Gammon-class neural net in a matter of a few days."

Unlike in chess, machines do not have an automatic advantage in backgammon. Because of the random element, the roll of the dice, every "ply" or set of moves has hundreds of possible outcomes, each of which, in turn, has hundreds more. The result is that even the fastest supercomputers can only look three or four moves ahead. The Deep Blue approach of searching through all possible moves and countermoves simply wouldn't work.

Instead, Tesauro's neural network chooses a move from all of those allowed based on its current dice throw, and that choice is allowed to propagate through the network. This is achieved by having every "neuron" in the network receive a signal, which it manipulates using a simple function and passes on to its neighbors. In reinforcement learning, when this signal reaches the output, it represents the network's prediction of the move's outcome. This is then compared with the actual consequence of making that move, and an error signal propagates back through the network.

The problem with a game like backgammon, as opposed to some sort of pattern-classification system, is that the real consequence of making moves-winning or losing-isn't actually known until the end.

The temporal-difference me-thod is designed to compensate for this, by forcing the error signal to propagate back in time. Not only does it adjust its interconnection weights-memory elements that store the machine's experience-for its last move, but it also looks at the weights for the previous moves. In effect, it looks back and attributes credit to all the moves that got it to its final destination. As one might expect, the error signal decreases as it goes back through time, so moves at the end of the game are much more highly "rewarded" or "penalized" than those at the beginning or middle of the game. This method of taking the timing of sequential moves into account effectively allows the network to discriminate between which helped it to win a game, and which caused it to lose.

Murray Campbell, a member of IBM's Deep Blue team, said he is impressed with the success of the new approach. "I think that it would be an interesting project to combine a neural-network evaluation function, as in TD-Gammon, with a searching program to create a chess-playing system," Campbell said. "There are clear difficulties here in terms of the amount of computation involved, but that is in part what makes the problem interesting." The strategy of having a machine "teach itself" to play a game is also appealing, he said. "Tesauro did a remarkable job in backgammon, but it seems to be more difficult in other games."

Though it did win all but one game at the AAAI tournament, TD-Gammon actually lost the match overall, thanks to a doubled win by then World Champion Malcolm Davis. This gave Davis a huge lead which, despite winning every game, the machine could never surmount. Ironically, it turns out that TD-Gammon does not use its neural network to decide whether to double, a method of upping the stakes of a game: it uses a conventional program instead.

Despite what could be called a close shave, Davis is a fan of TD-Gammon. "I have been playing backgammon for 22 years and always try to take advantage of the computer to learn," he said. "Occasionally the machine will find a play that almost all humans will overlook, and many times it will be right, as best we can tell." Plus, "when humans encounter adversity, it affects them, their play and their results in varying degrees. TD-Gammon is impervious to such emotional swings . . . Gerry Tesauro and IBM have changed the backgammon world dramatically."

Though Tesauro is pleased that his work has affected the game in this way, he is quick to point out that gaming is not the reason for building it. "I think it's a mistake to view TD-Gammon as simply an exercise in engineering a high-performance backgammon machine . . . rather, the goal was to use the game of backgammon as a test bed to develop powerful, flexible, general-purpose learning algorithms by which a machine can learn to perform any sequential decision-making task, simply from its own experience in attempting to perform the task. While I can't say that TD learning will work on any task," he said, "we have obtained quite a lot of understanding about types of tasks for which it will work."

More than a curiosity

Tom Dietterich, professor of computer zcience at Oregon State University in Corvallis, Ore., said Tesauro's impact will be far-reaching. "TD-Gammon was the first convincing application of reinforcement learning methods." Prior to that, he said, the technique was a laboratory curiosity. "People had done a lot of simulation work on simple problems, but no one had tackled a really large problem. Indeed, Gerry originally tried reinforcement learning methods expecting that they would fail miserably. His tremendous success was astonishing!" Dietterich himself had always regarded reinforcement learning as a method that could never handle real-world problems of practical size.

That situation has changed. "Gerry's success convinced me to give reinforcement learning another look," he said.

Copyright ® 1998 CMP Media Inc.

Sponsored Supplements From CMPnet

Win2000 ships in Oct: Codify your implementation plans now!

Eight steps to choosing the right Web hosting company for your business.

- Avoid technology triage with project management tools.
- How a major retailer is making Palm PCs a new sales generator.
- What to do when your company needs a portal of its own.
- Check out Apple's multi-colored PC savior, the iMac.

© 2001 CMP Media Inc.