Complex Problem Solving With Neural Networks: Learning Chess

Jack Sigan
Dr. Aleksander Malinowski, Advisor
Department of Electrical and Computer Engineering
Bradley University

Applied correctly and appropriately designed, artificial neural networks (ANNs) have extraordinary potential for solving problems involving generalization, trend recognition, and pattern matching. Game play, which often involves non-linear strategies or decision making, is a particularly good area to demonstrate the ANN as a way of approximating otherwise inexpressible functions [1]. To date, the promise and lofty expectations of this artificial intelligence approach have yet to be fully realized, demanding further research. Work on complex problem solving, such as that required in classical board games such as chess, has been limited, although many of the available research results [1-4] are tantalizing.

The purpose of this project is to develop an ANN which will learn and play the full game of chess as the dark side (whose first move follows light). Numerous published studies serve as motivation and a starting point for this research. Chekkapilla and Fogel, in developing an ANN to play checkers, indicated that there is feasibility in teaching ANNs games of some complexity [1,2]. Although chess is clearly a leap forward from checkers, it seems a logical next step in the evolution and development of the ANN, as chess is one of the most widely studied and researched games. A demonstration of strategy in such a game is also recognized as a direct measure of logical decision making ability [1]. Chess as an ANN problem is not a new idea; in fact, ANNs have already been proven to be highly effective in playing chess endgames [3]. The “Distributed Chess Project,” when considering the full game of chess, reported approximately 75% accuracy in choosing the “proper” move when confronted with a chess problem external to the training domain. While not fully successful as implemented, the study does appear to indicate that chess schema may be learned by ANNs [4].

By implementing a parallel architecture of partially connected feed-forward ANNs individually specialized in evaluating specific chess moves, it is hoped that the ANN will demonstrate a level of chess performance exceeding any study in this area known to be published; well enough to earn a rating of 1000 or above (USCF E Class). Each individual ANN will be structured from modular components thought to be conductive to schema development on the 8x8 board, and will contain features reminiscent of adaptive resonance [5] to ensure the “rules” of chess are obeyed at all times. If successful, the viability of the ANN as a method of developing complex strategy will become much more convincing. The modular nature of the individual ANNs proposed is especially appealing in that distributed processing could be used for training and evaluation. Training is also anticipated to be far more rapid than if a traditional feed-forward topology was employed.

Thus far, work towards the main research goal has concentrated on developing an ANN framework, including all processing, learning, and management functionality. Some time has also been spent determining the best way to store training records, including the data representation, as well as normalization and standardization considerations. Although work on the ANN platform is estimated to be roughly 80% complete at this point, the choice was made recently to discontinue work on the ANN framework and instead utilize the existing Stuttgart Neural Network Simulator (SNNS) with Java interface [6]. Work is now focusing on network topology design, network size considerations, and integrating the rule logic with the ANN components.

References:

[1] K. Chellapilla and D.B. Fogel. “Evolution, Neural Networks, Games and Intelligence.” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1471-1496, Sept. 1999.

[2] K. Chellapilla and D.B. Fogel. "Anaconda Defeats Hoyle 6-0: A Case Study Competing an Evolved Checkers Program Against Commercially Available Software,” in *Proceedings of the 2000 Congress on Evolutionary Computation*, 2000, vol. 2, pp. 857-863.

[3] C. Posthoff, S. Schawelski and M. Schlosser. “Neural Network Learning In a Chess Endgame,” in *IEEE World Congress on Computational Intelligence*, 1994, vol. 5, pp. 3420-3425.

[4] R. Seliger. “The Distributed Chess Project.” Internet: http://neural-chess.netfirms.com/HTML/project.html, 2003 [Aug. 26, 2004].

[5] M. Chester. *Neural Networks A Tutorial*. Englewood Cliffs, NJ: PTR Prentice Hall, 1993, pp. 71-81.

[6] University of Tübingen. “Stuttgart Neural Network Simulator.” Internet: http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html, 2003 [Aug 30, 2004].