Computer Science

COMPLEX PROBLEM SOLVING WITH NEURAL NETWORKS: LEARNING CHESS

Jack E. Sigan, Dr. Aleksander Malinowski*
Bradley University Department of Electrical and Computer Engineering
1501 West Bradley Avenue Peoria, IL 61625 Department Email: rsk@bradley.edu

Applied correctly and appropriately designed, artificial neural networks 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 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 artificial neural network which will learn and play the full game of chess as the dark side (whose first move follows light). Numerous published studies [2,3,4] serving as motivation for the project will be introduced briefly, with a focus on networks evolved from genetic algorithms, prior to discussing the current research status. Although the project specific discussion will focus on the current research direction, mention will be made of early development of an ANN framework prior to the decision to utilize the Stuttgart Neural Network Simulator [5].

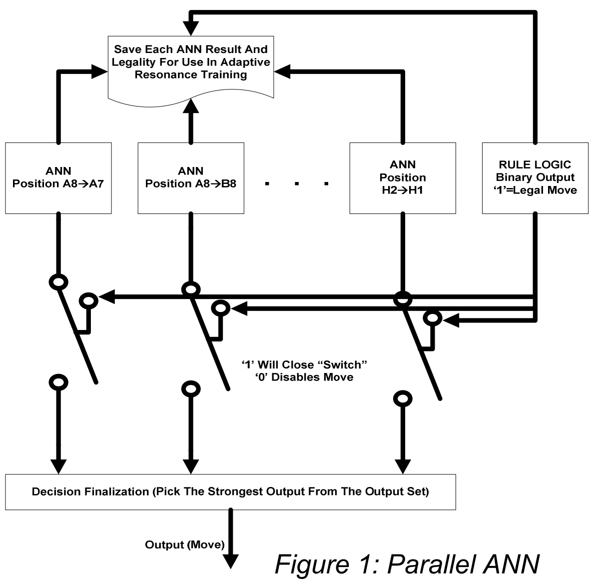

Figure 1: Parallel ANN Structure With Rule Logic

The main focus of the research, which is network topology design for a complex problem, will be discussed as being implemented for the specific problem of chess. A parallel architecture of partially connected feed-forward ANNs which are individually specialized in evaluating specific chess moves is the general concept under consideration (figure 1). The techniques used to ensure the "rules" of chess are obeyed at all times, reminiscent of adaptive resonance [6], will be outlined with the topology description. Training, although still in progress, will be briefly mentioned in terms of input vector creation, data sorting, and game preprocessing. An overview of the process used to convert a game record into input vectors will be shown. The costs and benefits of the modular nature of the network, including training time, processing speed, and of course decision accuracy will all be discussed.

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

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