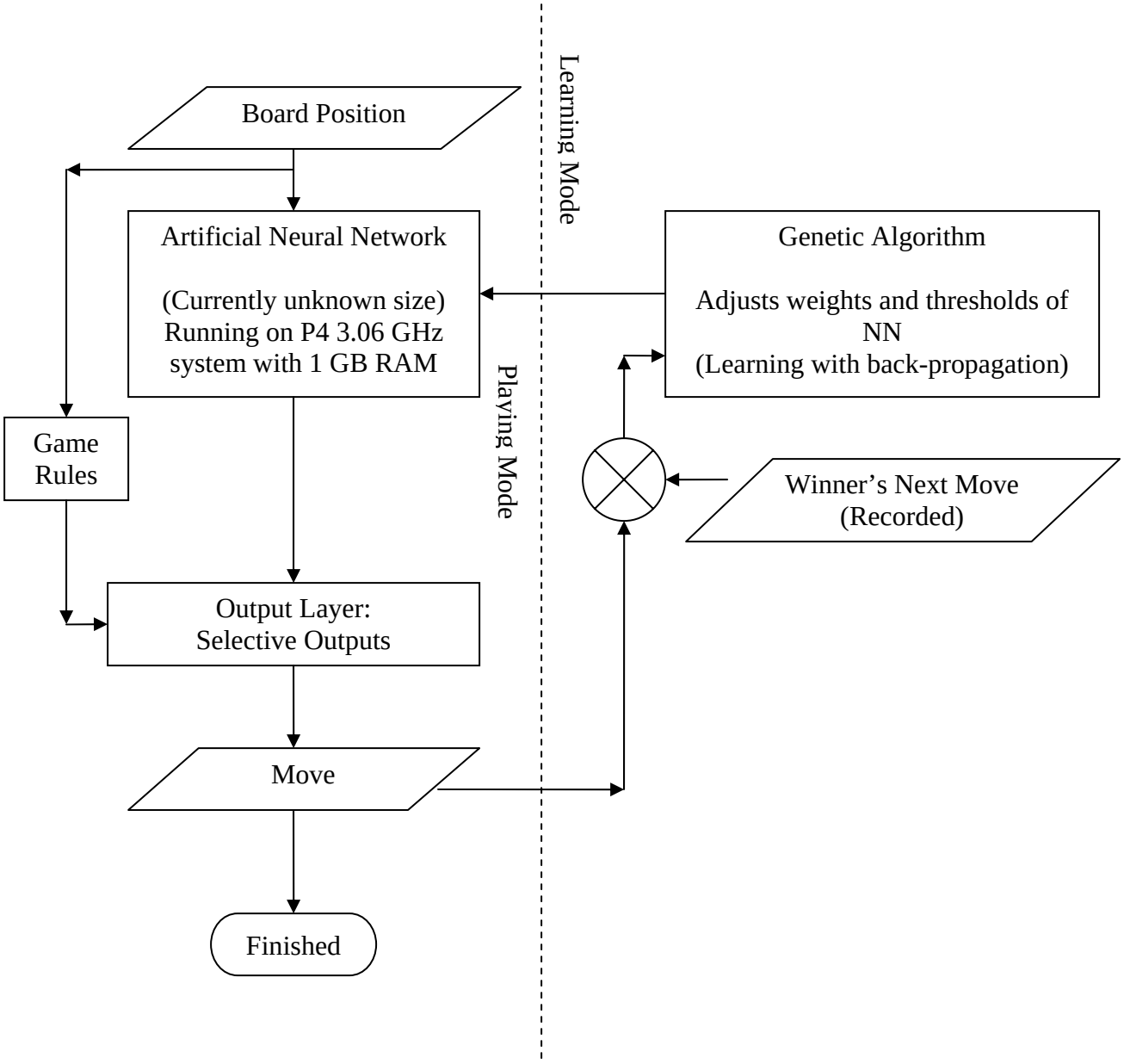
Jack Sigan
Bradley University ECE Dept
Dr. Malinowski Advisor
April 2004

Purpose and description:

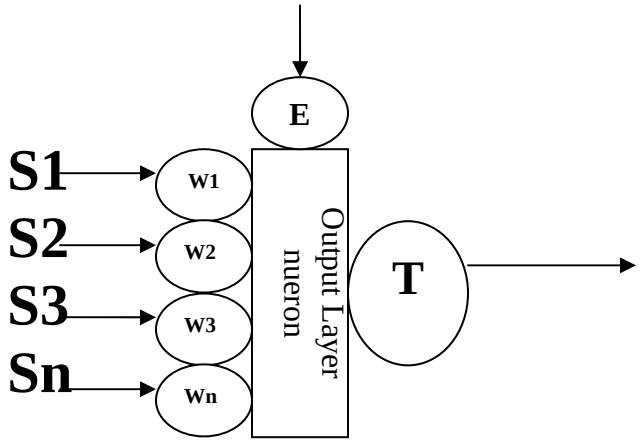
This project's purpose will be to create a system of artificial intelligence ultimately capable of playing chess against a human opponent (always playing as one side). The system will consist of an artificial neural network (ANN) with an input layer taking in data for the complete board position. Two modes, "learning" and "playing," will be implemented. The system block diagram for playing mode is shown on the left of figure 1 while the learning mode is shown on the right.

Figure 1: (System block diagram showing two modes of operation)



The output layer of the ANN will have connections to additional logic holding basic chess rules, to ensure that all moves fit within the constraints of the game. See figure 2 for a proposed output neuron model. By applying "1" or "0" to the E input of the neuron model, the output can be enabled or disabled. Each possible board position change (4032 of them in all) will have a dedicated output neuron. Moves which are illegal, invalid, or impossible by the rules of chess will have a value of 0 assigned to the E input, ensuring such moves are not made.

Figure 2: (Proposed output layer neuron model)



Output= E * (1/(1+e^{-ax}))
X = (S1W1+S2W2+S3W3+SnWn) – T
E = {0,1}
A > 0
T<W1+W2+W3…+Wn

The neural network will be taught the game through automated supervised learning with the input of potentially millions of recorded games (in "portable game notation" PGN files). Millions of recorded games (gigabytes worth) are widely available on the internet as archives, and merely need to be converted to a form suitable for ANN learning. The wide availability of sample data should offer a high potential to adequately train the network.

The system will learn through a system of back propagation with the application of a genetic algorithm. To explain this further, consider an output vector consisting of numerous floating point values between 0 and 1, representing possible chess moves. The move the ANN would like to make will be the highest value in the vector. To teach the network, the desired output will be forced to be the highest element in the vector by changing the weights and thresholds of selected neurons in the hidden and output layers.

The genetic algorithm will be used to find an appropriate value set which best accomplishes the following initial goals:

1. *The desired output must end up being the highest in the vector*
2. *Changes to other elements in the vector should be minimized*
3. *The number of neurons involved in the adjustment should be minimized*
4. *Weights should be adjusted before thresholds*
5. *Adjusting "a" to increase the sharpness of a Sigmoid function should be done as a last resort. See figure 2 for the location of the "a" variable.*

This list is not meant to be all inclusive, and is currently based only on intuitive concepts aimed at preventing data from being "un-learned." Essentially, the genetic algorithm will be solving systems of multivariable equations.

The output of the system will initially be a move in algebraic notation. It is also proposed to create a graphical user interface featuring a board and movable pieces so the game may be played in a fashion similar to "Yahoo Chess."