Tic Tac Toe Time

NEW MEXICO ADVENTURES IN

SUPERCOMPUTING CHALLENGE

FINAL REPORT

APRIL 8, 2005

TEAM 093A

Project Member:

Felicia Picknell

Teacher:

Debra Loftin

Project Mentor:

Nick Bennett

Table of Contents

Executive Summary

The purpose of this project is to create an artificial intelligence chess game opponent. It will be able perform the basic game movements of tic tac toe. For each move the program will counteract the move and make a response depending whether the previous move would lead towards the winning of the computer or the player. The game will eventually always end in a “tie” game.

Introduction

The significance of this project is to determine how to make a computer learn a strategy that will cause it to be more successful when playing a simple game. Games are the most common place where people are trying to implement “learned” responses. One example of using “learned” responses would be Neural Networks; they have many different ways of sorting out the “learned” data from each input.

Basis of Tic Tac Toe Memory

The memorisation of patterns and the subsequent response of the network can be categorised into two general paradigms:

associative mapping in which the network learns to produce a particular pattern on the set of input units whenever another particular pattern is applied on the set of input units. The associtive mapping can generally be broken down into two mechanisms:

auto-association: an input pattern is associated with itself and the states of input and output units coincide. This is used to provide pattern completition, ie to produce a pattern whenever a portion of it or a distorted pattern is presented. In the second case, the network actually stores pairs of patterns building an association between two sets of patterns.

hetero-association: is related to two recall mechanisms:

nearest-neighbour recall, where the output pattern produced corresponds to the input pattern stored, which is closest to the pattern presented, and

interpolative recall, where the output pattern is a similarity dependent

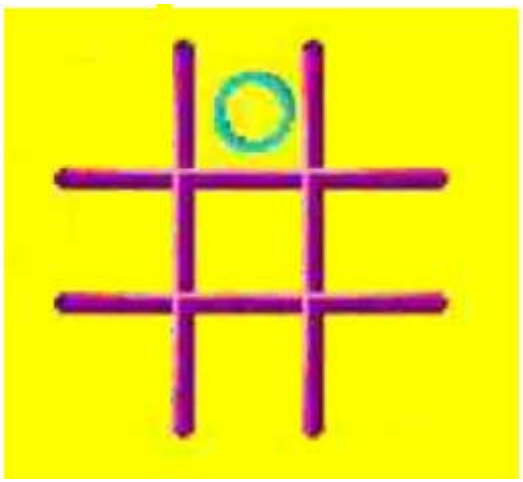
Every neural network posseses knowledge which is contained in the values of the connections weights. Modifying the knowledge stored in the network as a function of experience implies a learning rule for changing the values of the weights.

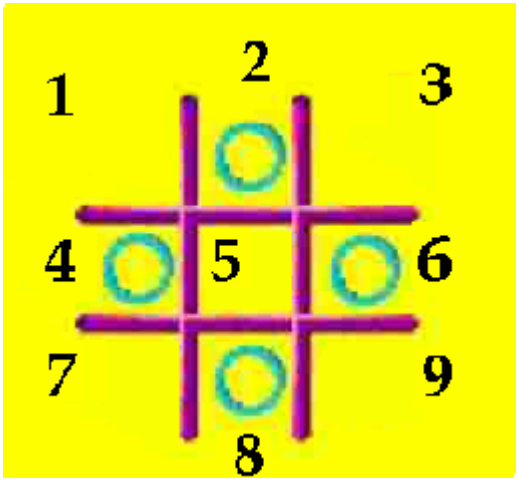
Move Composition List

The program's memory stores every game that is played. What I will have the computer do is, use every previous game and apply it to the current game being played. The computer will be able to use the previous information in about 3 different ways:

Storage Type One:

The Mirror Effect:





As you can see with the diagram above, with just one move the computer can see the move mirrored across the bored in all positions possible. If there were an X in one of the spots, such as square 8, then the computer would not mirror the circle in that square.

Acknowledgments

I would like to thank my whole family for pushing me through the project. I would also like to thank Ms. Loftin for helping me in many ways.

References

- http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html