Derek Lee Artificial Intelligence Fall 2020

Professor Sable Project #2

**Dataset**

I used the Tic-Tac-Toe Endgame Dataset from the UCI ML Repository (<http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame>). This is a relatively simple dataset. It “encodes the complete set of possible board configurations at the end of tic-tac-toe games, where ‘x’ is assumed to have played first.”

It has 9 input attributes, each representing 1 square on the board. Each input attribute is an element from the set (x, o, b). x represents an x on the square, o represents an o on the square, and b represents a blank square.

It has 1 output attribute, representing whether x won the game. Positive represents a victory for x, while Negative represents a loss for x.

**Parameters**

I found good results with:

* Learning rate of 0.1
* 25 hidden nodes
* 100 epochs

**Initial Weights**

I used the same approach as the one for the WDBC dataset and the grades dataset; I pseudo-randomly generated fractions between 0 and 1. The file used is randomNetwork.cpp.

**Modifications**

The files used for modifying the initial dataset are preprocessing.cpp and splitData.cpp. preprocessing.cpp was used to perform the modifications specified below. splitData.cpp was used to split the modified dataset into the training and testing sets.

I converted x to 1, o to -1, and b to 0. I realized that using a dataset with continuous rather than discrete attributes may have given me better results, but I thought this dataset was interesting. I figured that representing an X with a 1, an O with a -1, and a B with a 0 would be a logical transformation of the input attributes. An X represents the opposite of an O, while a B is neutral.

I converted Positive to 1, and Negative to 0. My logic is that Positive is desired, and thus should be represented as a True bool value, but I do not think this decision had a significant effect. I should be able to get the same results by converting Positive to 0 and Negative to 1.