Universidad de Buenos Aires
Facultad De Ingeniería
Año 2012 - 1er Cuatrimestre

Sistema de Programación No Convencional de Robots (75.70)

Trabajo Práctico Final
El juego Ta-Te-Ti

Integrantes

Apellido, Nombre	**Nro. Padrón**	**E-mail**
Bukaczewski, Verónica	86954	vero13@gmail.com
Rivero, Hernán	88455	riverohernanj@gmail.com

Índice

1. Objetivo

El objetivo del presente trabajo práctico es entrenar una red neuronal para que juegue al Ta-Te-Ti. Se utilizará una red neuronal de tipo backpropagation, para generar un método de aprendizaje a medida que se desarrollan las partidas de Ta-Te-Ti.

2. El juego Ta-Te-Ti

Por lo general, el Ta-Te-Ti se juega en una cuadrícula de tres por tres (ver Figura [1]). Cada jugador, a su vez se mueve mediante la colocación de un marcador en un casillero vacío. El marcador de un jugador es "X"(cruz) y el del otro es "O"(círculo). El juego termina tan pronto como un jugador tiene tres marcadores en una fila: horizontalmente, verticalmente, o en diagonal (un ejemplo se muestra en la Figura [2]). El juego puede también terminan en empate (ver Figura [3]), si no hay posibilidad de ganar para alguno de los jugadores.

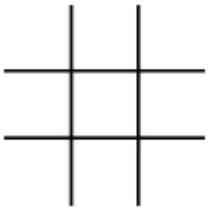

Figura 1: Grilla vacía TaTeTi.

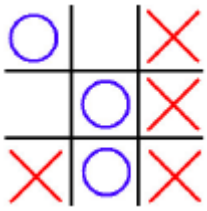

Figura 2: El jugador cruz gana la partida.

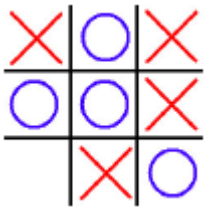

Figura 3: La partida terminó empatada.

3. Estructura general de la solución propuesta

Se realizó un programa en lenguaje Java y se utilizó el framework de red neuronal Joone. A continuación se presenta el diagrama de clases.

4. Soluciones propuestas

Se armaron distintas estructuras de redes neuronales, con el objetivo de poder entrenarlas para que jueguen al Ta-Te-Ti. El primer intento fue la realización de una red con nueve inputs y nueve outputs, al cual se le variaron la cantidad de capas ocultas. Los inputs representan los nueve casilleros del tablero y se especifican con:

- 1: ficha de la red neuronal.
- -1: ficha rival.
- 0: casillero vacío.

Los outputs representan los nueve casillero del tablero, pero esta vez se le asocia un puntaje que nos indica que tan apropiado es jugar en dicho casillero.
Luego de varios intentos de aprendizaje, quedó descartada porque no mostraba cambios significativos al variar los inputs.

5. Solución utilizada

5.1. Estructura de la Red Neuronal

Se armó una red con dieciocho valores de input, organizados de la siguiente manera:

- del 1 al 9 valen 1 si la posición está ocupada con una ficha de la red, 0 en caso contrario.
- del 10 al 18, valen 1 si hay una ficha del rival, 0 en caso contrario.

Los outputs son nueve y representan los puntajes para los casilleros del tablero. Se utilizaron tres capas ocultas de 18-27-27 filas. Se utilizaron estos número teniendo en cuenta la cantidad de casilleros del tablero de Ta-Te-Ti es 9 y que por cada uno se tiene la posibilidad de encontrar tres tipos de elementos (cruz, c´rculo y vacío).

5.2. Entrenando la Red Neuronal

El entrenamiento de la red neuronal se realizó mediante el juego contra otros algoritmos que juegan Ta-Te-Ti. Con el avance de las partidas, se le fue corrigiendo a la red neuronal donde debía jugar; para ellos se utilizó un sistema de puntajes de acuerdo a si ganó o perdió. De esta forma se le fue marcando un camino y con el pasar de las partidas, la red iba adquiriendo una mayor inteligencia a la hora de realizar sus jugadas.

5.3. Resultados

6. Conclusiones

Una primera conclusión nos indica que el armado de una red neuronal lleva un tiempo considerablemente mayor al del entrenamiento de la misma. Esto se debe a que el entrenamiento es un proceso automatizado, que no requiere más que la conguración de algunos parámetros. Mientras que para el armado de una red neuronal, que nos permita resolver el problema, es necesario un conocimiento sobre la situación que estamos enfrentando, de manera de poder crear una red que nos permita resolver el problema de manera eciente. Por otro lado, observamos que el entrenamiento prolongado de una red mal armada no signicaba llegar a los resultados deseados. Es decir, que lo más importante es concentrarnos en crear una red acorde al problema; de manera de asegurarnos llegar a los resultados buscados; lo cual puede estar acompañado de un tiempo de entrenamiento más corto de lo esperado.

ESCRIBIR ALGO SOBRE RESULTADOS OBTENIDOSSSSSSS

Referencias

[1] **Documentación Joone**
http://sourceforge.net/projects/joone/files/Documentation/DTE/JooneDTEGuide.pdf

[2] **Tutorial Básico Joone**
http://ubuntuone.com/p/1dB/

[3] **Training an artificial neuronal network to play tic-tac-toe**
http://users.auth.gr/kehagiat/GameTheory/12CombBiblio/TicTacToe.pdf

[4] **How to code an artificial neural network (Tic-tac-toe)?**
http://stackoverflow.com/questions/761216/how-to-code-an-artificial-neural-network-tic-tac-toe

[5] **Neural Net Training for Tic-Tac-Toe**
www.cs.virginia.edu/~bmb5v/cs660/Project.doc

[6] **TD Learning of Game Evaluation Functions with Hierarchical Neural Architectures**
http://webber.physik.uni-freiburg.de/~hon/vorlss02/Literatur/reinforcement/GameEvaluationWithNeuronal.pdf