



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2012 - 2^{do} Cuatrimestre

SISTEMA DE PROGRAMACIÓN NO CONVENCIONAL DE ROBOTS
(75.70)

TRABAJO PRÁCTICO

Integrantes

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

Índice

1. Objetivo	3
2. Descripción base de datos seleccionada	3
2.1. Información relevante	3
3. Preparando los datos para las corridas	3
4. Red Neuronal	4
5. Ejecutando la Red Neuronal	5
6. Resultados	5
7. Conclusiones	6
A. Tabla comparativa de los resultados	7

1. Objetivo

El objetivo del presente trabajo práctico es familiarizarnos con la herramienta Joone, utilizada para el estudio de Redes Neuronales. Y finalmente, poder realizar una análisis de los resultados obtenidos.

2. Descripción base de datos seleccionada

Se seleccionó la base de datos del TA-TE-TI, extraída de la página UCI (Machine Learning Repository) [3]. Esta base de datos codifica el conjunto completo de configuraciones posibles para el final del juegos del TA-TE-TI, donde "x" se supone que juega primero. El concepto objetivo es "ganar para x" (es decir, ocurre cuando "x" tiene una de las 8 posibles maneras de crear un "tres-en-línea").

2.1. Información relevante

- Número de instancias: 958.
- Número de atributos: 10.
- Información de los atributos: (x=player x has taken, o=player o has taken, b=blank)
 1. top-left-square: x,o,b
 2. top-middle-square: x,o,b
 3. top-right-square: x,o,b
 4. middle-left-square: x,o,b
 5. middle-middle-square: x,o,b
 6. middle-right-square: x,o,b
 7. bottom-left-square: x,o,b
 8. bottom-middle-square: x,o,b
 9. bottom-right-square: x,o,b
 10. Class: positive,negative
- Falta de valores de atributo: Ninguno.
- Distribución de Clase: 65,3 % son positivos (es decir, gana para "x").

3. Preparando los datos para las corridas

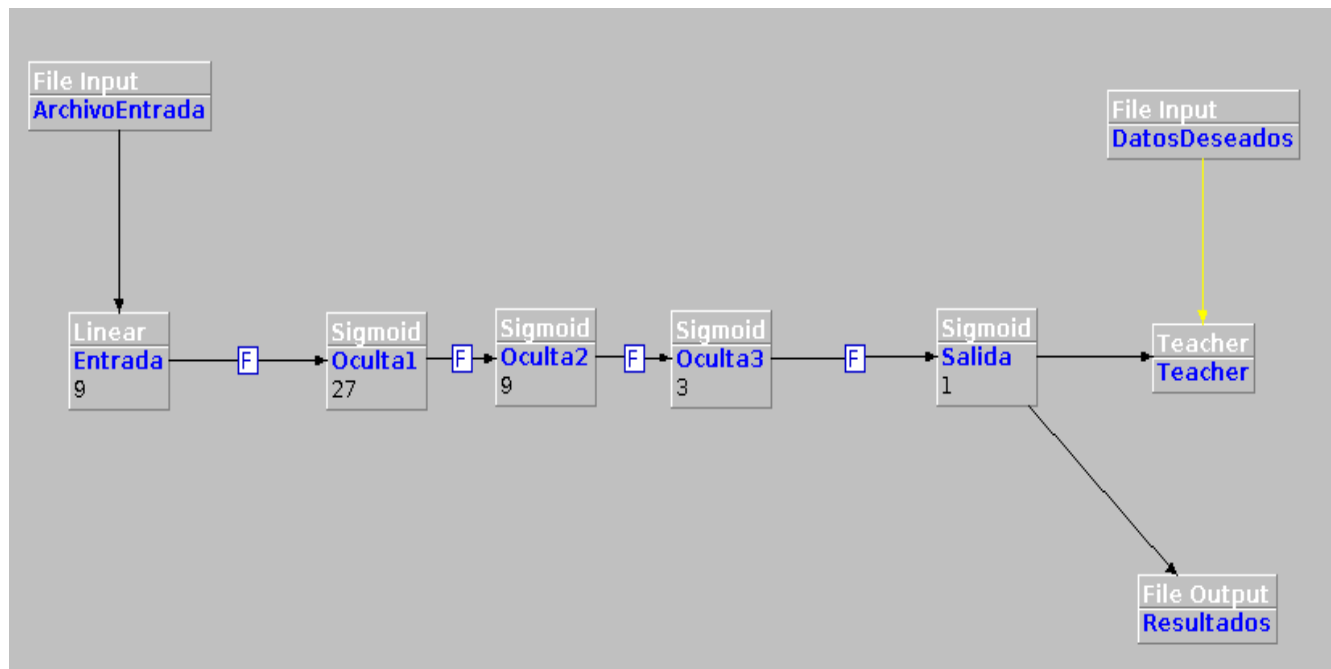
Los valores para los atributos fueron modificados para que el programa Joone pueda ejecutarse correctamente; debido a que sólo trabaja con números reales y enteros.

Valores:

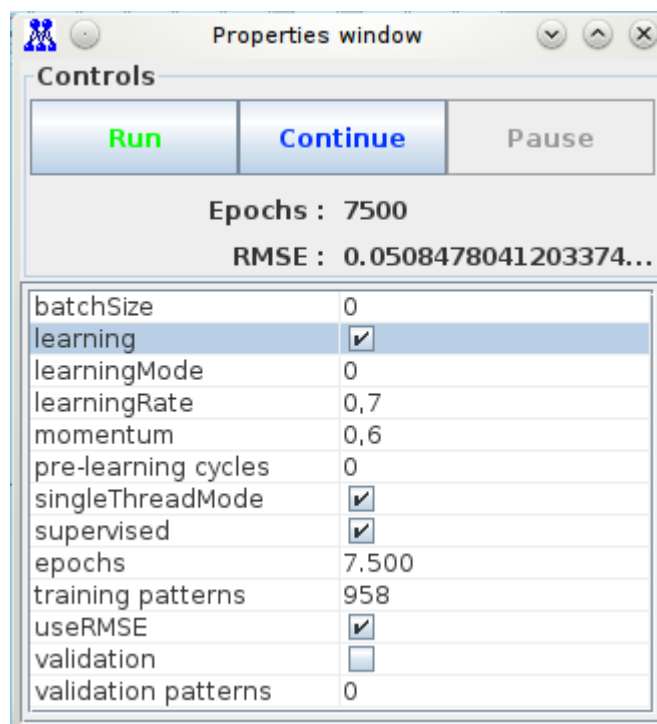
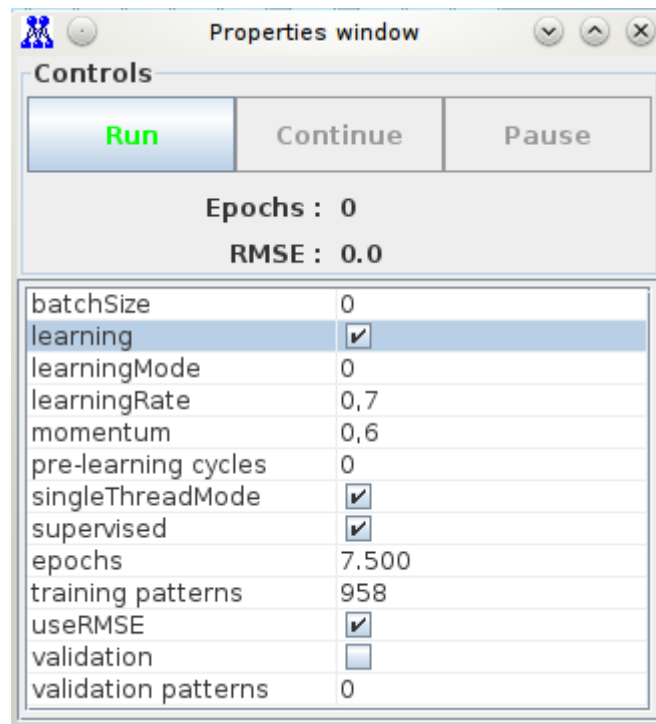
1. x= +1

2. $o = -1$
3. $b = 0$
4. $\text{positive} = 1$
5. $\text{negative} = 0$

4. Red Neuronal



5. Ejecutando la Red Neuronal



6. Resultados

Luego del aprendizaje que se le aplicó a la Red Neuronal, se agregó un archivo de salida para probar la red entrenada ("Resultados.txt"). Para ello, se configuró:

- learning = FALSE
- epochs = 1

En el archivo se pudo observar que los valores coinciden aproximadamente con la columna diez de la base de datos original.

7. Conclusiones

Cuanto lleva armarlo y cuando lleva correrlo.

A. Tabla comparativa de los resultados

A continuación se presentan los resultados originales de la base de datos contra los obtenidos de la red neuronal entrenada.

Referencias

- [1] **Documentación Joone**
<http://sourceforge.net/projects/joone/files/Documentation/DTE/JooneDTEGuide.pdf>
- [2] **Tutorial Básico Joone**
<http://ubuntuone.com/p/1dB/>
- [3] **UCI Machine Learning Repository - Tic-Tac-Toe Endgame Data Set**
<http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame>
- [4] **Training an artificial neuronal network to play tic-tac-toe**
<http://users.auth.gr/kehagiat/GameTheory/12CombBiblio/TicTacToe.pdf>
- [5] **How to code an artificial neural network (Tic-tac-toe)?**
<http://stackoverflow.com/questions/761216/how-to-code-an-artificial-neural-net>
- [6] **Neural Net Training for Tic-Tac-Toe**
www.cs.virginia.edu/~bmb5v/cs660/Project.doc
- [7] **TD Learning of Game Evaluation Functions with Hierarchical Neural Architectures**
<http://webber.physik.uni-freiburg.de/~hon/vorlss02/Literatur/reinforcement/GameEvaluationWithNeuronal.pdf>