



**Escuela de Ingeniería en Computación**

**Inteligencia Artificial**

**Prof. Luis Karlos Hernandez**

# **PROYECTO 3**

## **Redes Neuronales**

**Giacomo Ubaldo Pino - 2014024410**

**I Semestre, 2017**

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>2</b>
<b>OBJETIVOS</b>	<b>3</b>
OBJETIVO GENERAL	4
OBJETIVOS ESPECÍFICOS	4
<b>PROPUESTA</b>	<b>4</b>
<b>ANÁLISIS DE RESULTADOS</b>	<b>7</b>
<b>CONCLUSIONES</b>	<b>8</b>
<b>BIBLIOGRAFÍA</b>	<b>8</b>

## INTRODUCCIÓN

Las redes neuronales son un modelo computacional basado en un gran conjunto de unidades neuronales simples, de forma similar a las neuronas de los cerebros biológicos. Todas las unidades se conectan entre sí para modificar el estado de cada una.

La propagación hacia atrás es donde se utiliza la estimulación hacia adelante para restablecer los pesos de las unidades neuronales y esto a veces se realiza en combinación con una formación en la que se conoce el resultado correcto.

El objetivo de las redes neuronales es resolver problemas de la misma manera que los cerebros biológicos, siguiendo patrones y aprendiendo a dar resultados de acuerdo a entradas específicas.

Este proyecto propone una aplicación de las redes neuronales al segundo proyecto del curso de Inteligencia Artificial, en el cual se simulaban taxis al rededor de una ciudad.

## **OBJETIVOS**

En la presente sección se especifica el objetivo general de la realización general de este proyecto, así como también, cada objetivo específico que se ha tenido en

cuenta en el marco de desarrollo de este trabajo.

## **OBJETIVO GENERAL**

- Proponer una solución al problema de enseñar a un taxi a conducir por la ciudad haciendo uso de redes neuronales

## **OBJETIVOS ESPECÍFICOS**

- Implementar una clase para la aplicación de redes neuronales
- Modificar el proyecto 2 para adaptar la clase de redes neuronales a este.
- Implementar nuevas funcionalidades para conectar ambas partes.

## **PROPUESTA**

A continuación se describe la propuesta hecha en este trabajo para hacer uso de las redes neuronales en el proyecto 2 de Inteligencia Artificial:

### **Consideraciones**

- Solo se tomará en cuenta el uso de un taxi.
- No existirán clientes durante esta simulación, por lo cual tampoco hay funcionalidades de recogerlos y llevarlos a un destino.
- No se aplicará el algoritmo A\*.
- El mapa tendrá que ser de forma rectangular o cuadrangular.
- Lo que se espera del taxi es que se pueda mover libremente por la ciudad sin chocar con obstáculos.

## Descripción

### - Entrada (Input layer):

La red neuronal propuesta recibirá 736 entradas, este número es la cantidad de caracteres que existen en el archivo del mapa original. La entrada entonces consiste del mapa en sí.

### - Capa intermedia (Hidden layer):

Los nodos de la capa intermedia serán 200, esto elegido arbitrariamente pero manteniéndose entre el rango de la cantidad de nodos de entrada y los de salida.

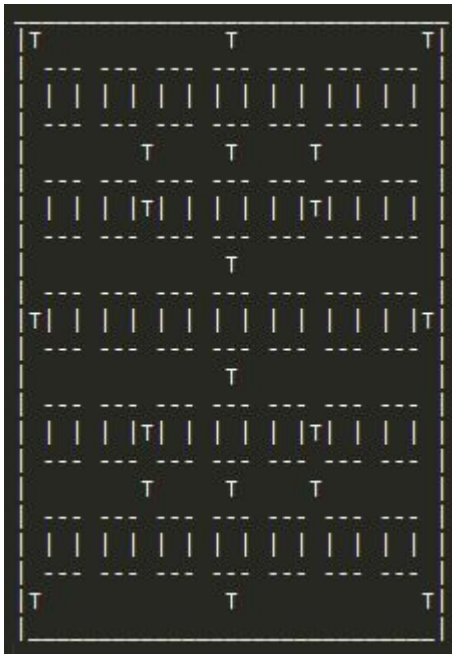
### - Salida (Output layer):

La cantidad de nodos de salida será de 4. Cada uno representa una dirección del taxi a tomar: Arriba, Derecha, Abajo, Izquierda.

De esta manera, la respuesta de la red neuronal a un mapa dado como entrada, con el taxi en una posición específica, será de una dirección a donde ir después.

## Entrenamiento

Para entrenar a la red neuronal se utilizaron mapas que cubrieran posiciones del taxi diversas, con diferentes obstáculos en diferentes posiciones adyacentes al taxi. El esquema que se siguió fue el siguiente:



De esta manera, cada mapa cubría una posición del taxi. Como se puede observar, cada posición puede tener como opción moverse a más de una dirección. Lo que se hizo entonces es repetir cada entrada tantas veces como posibles movimientos tenga, y se hizo una salida objetivo (target) para cada dirección que pueda tomar.

Las entradas se guardaron en un formato csv en donde cada línea era un mapa y consistía de los 736 caracteres. Cada uno de estos caracteres fue representado con un número en base a 255, todos los obstáculos fueron representados con 127, los espacios en blanco con 0, siendo el mínimo y el taxi con 255, siendo el máximo (1).

Se realizó una cantidad de 1 entrenamiento con las entradas y targets. La red neuronal se creó con un ratio de aprendizaje de 0.1.

### **Aplicación dentro del programa**

Como código base se tomó la implementación del taxi en la ciudad del proyecto 2. A este se le hicieron las siguientes modificaciones:

- Agregación de una clase RedNeuronal que incluía el constructor, la función

de entrenamiento y la de llamada de uso.

- Agregación de funciones para permitir la lectura de los archivos para el entrenamiento de la red neuronal.
- Cambio de llamada para la funcionalidad del movimiento de taxis. Para que así el cambio de posición de un taxi sea de acuerdo a la propuesta dada por la red neuronal.

## **ANÁLISIS DE RESULTADOS**

Después de correr el programa, se puede observar que el taxi solo sigue una misma dirección (derecha) sin cambio alguno, hasta llegar al extremo y chocar con el obstáculo. Este resultado se dio con 1 entrenamiento de las entradas y targets descritos.

Al intentar modificar el número de entrenamientos, el resultado varía pero negativamente. Por ejemplo, después de aumentar a 10 entrenamientos de las entradas y targets descritos anteriormente, el taxi, que comienza en la esquina superior izquierda, toma como elección irse más a la izquierda, chocando con un obstáculo inmediatamente al iniciar. Al aumentar a 100 entrenamientos, se sigue obteniendo este mismo resultado.

Se intentó modificar el número de nodos en la capa intermedia, pero no genera cambios en los resultados anteriores. Finalmente, se aumentó el ratio de aprendizaje a 1; y, con un entrenamiento, el taxi tomó una dirección hacia abajo, y siguió esta misma hasta llegar al obstáculo del extremo inferior.

## **CONCLUSIONES**

Se puede concluir que el comportamiento de la red neuronal creada es bastante inestable, pues brinda direcciones distintas con cada corrida del programa, pero estas se mantienen constantes y no tienen mucho sentido. Esta conducta se mantiene independientemente del número de nodos en la capa intermedia, del ratio de aprendizaje, y del número de entrenamientos.

La solución propuesta de este punto es modificar las entradas y salidas objetivo (targets), de una forma más específica o más estratégica, para así lograr que la red neuronal proponga movimientos coherentes y cambiantes, tal y como lo haría un taxi de verdad.

Por último, se concluye que las redes neuronales sí son bastante útiles a la hora de resolver problemas, si estas están bien planteadas y se entrenan de manera óptima. Asimismo, su uso e implementación son bastante simples, por lo que se prestan a la solución de múltiples tipos de problemas.

## **BIBLIOGRAFÍA**

A Neural Network in 11 lines of Python (Part 1). Recuperado el 19/06/2017 de <https://iamtrask.github.io/2015/07/12/basic-python-network/>



