Back propagation

Set of segments (Back propagation).

Autores: Cristian Andrés Arce Pineda, Juan Pablo Pérez Santos.

*Computación Blanda, Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira.*

Correo-e: [cristian.arce@utp.edu.co](mailto:cristian.arce@utp.edu.co),[juanpablo.perez@utp.edu.co](mailto:juanpablo.perez@utp.edu.co).

***Resumen*— En el siguiente documento se explicará el desarrollo y el funcionamiento de un juego de segmentos implementado un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales, llamado Backpropagation.**

***Palabras clave—*. Neurona, red, algoritmo, señal, pesos, reconocer, clasificar, agente, inteligente, búsqueda.**

***Abstract*— - The following document will explain the development and operation of a set of segments implemented a gradient calculation method used in supervised learning algorithms used to train artificial neural networks, called Backpropagation.**

***Key Word* —. Neuron, network, algorithm, signal, weights, recognize, classify, agent, intelligent, search.**

1. INTRODUCCIÓN

La computación blanda busca integrar diferentes paradigmas computacionales, como las redes neuronales, la lógica difusa y los algoritmos genéticos, donde cada una de ellas aparentan ser muy efectivas en la manipulación de datos dinámicos, no lineales y ruidosos. Sin embargo, cuando se utilizan conjuntamente se aprovecha la fortaleza de cada una de forma sinérgica en el desarrollo de sistemas híbridos. Las redes neuronales son similares a la regresión no lineal, pero son más robustas y pueden detectar relaciones ocultas en grandes conjuntos de datos utilizando la teoría de reconocimiento de patrones.

De esta forma se puede implementar en un visualizador de siete segmentos que permite observar números del 0 al 9, permitiendo reconocer cual es el número que se está ingresando por medio del teclado

1. CONTENIDO

**1. Herramientas utilizadas.**

Para el desarrollo de este programa se utilizó el lenguaje de programación PYTHON, la librería PYGAME que permite realizar una interfaz gráfica la cual permite observar el funcionamiento del programa de manera más organizada.

**2. Desarrollo del programa.**

En primer lugar, el programa consiste en un juego de 7 segmentos, que simula un reloj digital y que muestra cualquier tipo de número del 0 al 9, se decidió comenzar a desarrollar este programa con 3 capas, capa de entrada, capa oculta y capa de salida. La asignación del numero de neuronas para cada capa fue propuesta de la siguiente manera:

* Capa de entrada: 7 neuronas.
* Capa oculta: 7 neuronas.
* Capa de salida: 4 neuronas.

Para la capa de salida se definieron utilizar 4 neuronas ya que estas simbolizan el numero de bits necesarios para representar un número que este dentro del rango del 0 al 9.

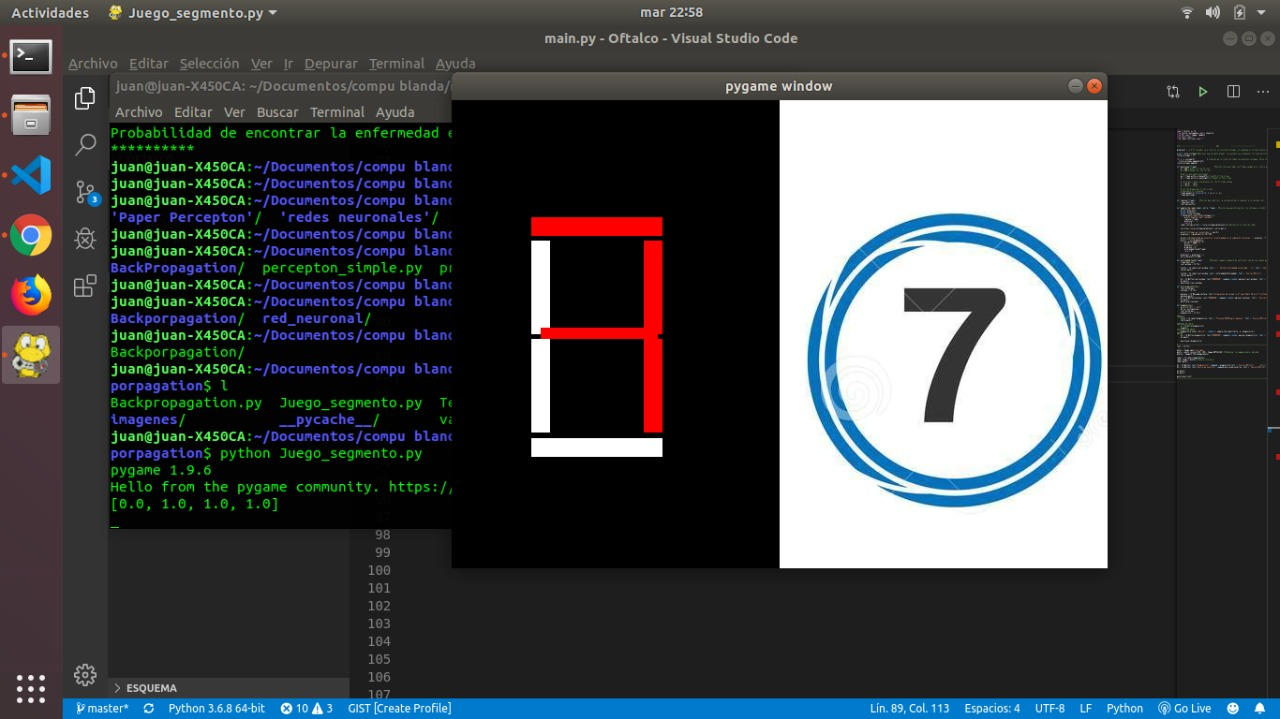


Figura 1. Ejemplo imagen

El funcionamiento del programa se divide en:

1. Se elegirá el tipo de número que se quiere visualizar en él panel.
2. El programa captura los datos introducidos.
3. Creación de vector.
4. Respuesta del programa.

**Elegir el tipo de numero a ingresar:** La primera opción que nos permitirá este programa será ingresar el numero que queremos que nos reconozca nuestra red neuronal.

**El programa captura los datos introducidos:** Después de haber ingresado el número, cada segmento seleccionado por el usuario cambiará de tonalidad de blanco a rojo y con la tecla espacio se enviará el numero al algoritmo para que sea capturado y se comience hacer su debido tratamiento.

**Creación de vector:** Después de que el programa haya capturado los datos suministrados por el usuario, cada segmento en rojo se convertirá en un uno (1) y los que no estén activos serán cero (0), de esta forma se crea un vector, que se convierte en la entrada para la red neuronal pueda darnos por medio de la capa de salida la respuesta correcta.

**Respuesta del programa:** Para finalizar, después de tener el vector con las respectivas salidas de cada segmento la red neuronal se encargará de obtener el número ingresado por el usuario.

**RESULTADO.**

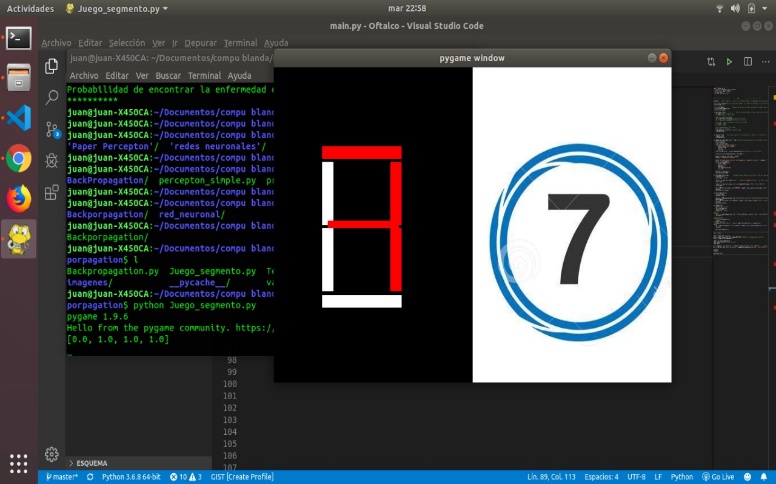


Figura 1.2. Ejemplo de gráfico.

1. CONCLUSIONES

Del anterior desarrollo se concluye:

* Las redes neuronales tienen un potencial muy interesante en la implementación de este tipo de ejercicios.
* Aprendimos de qué forma se puede implementar backpropagation y de esta manera saberlo utilizar para el futuro en algún proyecto que lo requiera.

REFERENCIAS

1. Computacion blanda definición. Recuperado de: <http://computacionblanda.blogspot.com/>
2. Pygame . Recuperado de:

<https://matplotlib.org/>

[3] Backpropagation. Recuperado de: <https://es.wikipedia.org/wiki/Propagaci%C3%B3n_hacia_atr%C3%A1s>