

# MANUAL DE USUARIO DE UNA SIMULACIÓN DE UNA RED NEURONAL A TRAVÉS DE GRAFOS

## [DESCRIPCIÓN GENERAL](#)

### [TÉRMINOS CLAVES](#)

[Nodo](#)

[Topología](#)

[Capa](#)

[Arista o conexión](#)

## [INGRESO DE INFORMACIÓN POR MEDIO DE ARCHIVOS](#)

### [ARCHIVO ESTRUCTURA DE LA RED](#)

### [DATOS DE ENTRENAMIENTO](#)

## [INGRESO DE INFORMACIÓN POR MEDIO DE COMANDOS](#)

### [ESTRUCTURA DE LA RED](#)

[Hasta este momento solo se ha introducido la forma de la red, a continuación introduciremos los datos de entrenamiento.](#)

### [DATOS DE ENTRENAMIENTO](#)

### [ENTRENAMIENTO](#)

## [EJEMPLO DE USO](#)

### [DEFINICIÓN DE LA RED](#)

### [DATOS DE PRUEBA](#)

### [EJECUCIÓN DEL PROGRAMA](#)

## [INTERPRETACIÓN DE LA SALIDA](#)

### [TÉRMINOS CLAVE](#)

[ITERACIÓN](#)

[RECORRIDO](#)

[EDICION](#)

## [DISPONIBILIDAD](#)

## [LICENCIA](#)

## DESCRIPCIÓN GENERAL

Esta herramienta permite la creación y entrenamiento de redes neuronales, actualmente existen varias herramientas para hacer este tipo de redes (por ejemplo, matlab) pero esta en específica crea y entrena la red mediante Grafos.

## TÉRMINOS CLAVES

### Nodo

Contiene el valor del umbral el cual. la suma de sus entradas multiplicado por las conexiones de las mismas debe ser mayor a la del umbral para que el valor de salida del nodo sea 1.

### Topología

Describe el tamaño de la red, su número de capas, su número de entradas y su número de salidas

### Capa

Está compuesta de 1 a 5 nodos y solo tiene conexiones hacia la siguiente capa.

### Arista o conexión

Se encarga de conectar un nodo con otro, tiene asociado un peso.

## INGRESO DE INFORMACIÓN POR MEDIO DE ARCHIVOS

Este programa provee la opción de introducir la estructura de la red mediante un archivo de texto, para hacer esto en el menú principal del programa seleccione la opción A. Una vez seleccionada la opción el programa le solicitará que indique la ruta donde tiene los archivos de entrenamiento y de descripción de la red.

A continuación se describe la estructura de los archivos de entrada.

### ARCHIVO ESTRUCTURA DE LA RED

En la **primera línea** de este archivo debe ir la topología de la red, esta compuesta de n elementos que indican el número de nodos por capa. Por ejemplo si tiene un grafo con 2 entradas, 2 capas intermedias y 1 capa de salida esta línea debe ser de la siguiente manera:

2 2 2 1

En la **segunda línea** deben ir los pesos o los umbrales de los nodos intermedios, en este caso desde la entrada hasta la salida hay 5 nodos (4 intermedios y 1 de salida), para este ejemplo se usarán los siguientes umbrales:

-0.5 -0.2 -0.2 -0.5 -0.3

En las siguientes líneas del archivo se define el peso de las conexiones entre los nodos separadas por espacio, si no existe una conexión entre estos se pondrá como valor 0.

0.0 0.0 -0.7 -0.7 0.0 0.0 0.0

0.0 0.0 0.2 0.2 0.0 0.0 0.0

-0.7 0.2 0.0 0.0 0.3 0.3 0.0

-0.7 0.2 0.0 0.0 -0.5 0.5 0.0

0.0 0.0 0.3 -0.5 0.0 0.0 0.2

0.0 0.0 0.3 0.5 0.0 0.0 0.5

0.0 0.0 0.0 0.0 0.2 0.5 0.0

## DATOS DE ENTRENAMIENTO

El archivo de entrenamiento está dado por N columnas, separadas por espacio donde:

N = Número de entradas + Número de salidas.

Por cada caso existe una nueva línea.

Para este caso en específico (se tienen 2 entradas y 1 salida) usaremos un archivo de entrenamiento con un compuerta OR, por lo tanto el archivo OR contiene la siguiente información:

0 0 0

0 1 1

1 0 1

1 1 1

## INGRESO DE INFORMACIÓN POR MEDIO DE COMANDOS

Para introducir la información por medio de la línea de comandos es necesario seleccionar en el menú principal la opción B, el procedimiento es similar al que se hizo a la hora de crear un archivo de entrada, la única diferencia es que el programa preguntará por cada uno de los componentes.

## ESTRUCTURA DE LA RED

Lo primero que el programa pedirá introducir es la topología de la red, para hacer esto usted debe introducir toda la topología en **una sola línea**.

2 2 2 1

Posteriormente el programa le preguntará por los pesos de los nodos, al igual que en el caso anterior debe introducir los pesos de los nodos en **una sola línea**.

-0.5 -0.2 -0.2 -0.5 -0.3

Luego de esto el programa calculará la cantidad de nodos que tiene la red y solicitará a ud introducir las conexiones entre ellos, en este caso usted introducirá las siguientes líneas una por una.

0.0 0.0 -0.7 -0.7 0.0 0.0 0.0

0.0 0.0 0.2 0.2 0.0 0.0 0.0

-0.7 0.2 0.0 0.0 0.3 0.3 0.0

-0.7 0.2 0.0 0.0 -0.5 0.5 0.0

0.0 0.0 0.3 -0.5 0.0 0.0 0.2

0.0 0.0 0.3 0.5 0.0 0.0 0.5

0.0 0.0 0.0 0.0 0.2 0.5 0.0

Hasta este momento solo se ha introducido la forma de la red, a continuación introduciremos los datos de entrenamiento.

## DATOS DE ENTRENAMIENTO

Para introducir estos datos el programa solicitará a usted inicialmente el número de casos a probar para hacer el entrenamiento, en este seleccionaremos 4 ya que trabajaremos con la compuerta OR.

4

Luego de esto el programa le solicitará que introduzca información 4 veces (una por caso), usted debe suministrar al programa uno por uno los casos de entrada y salida.

0 0 0

0 1 1

1 0 1

1 1 1

## ENTRENAMIENTO

Una vez haya cargado los datos puede empezar a hacer el entrenamiento del grafo seleccionando la opción C del menú principal

## EJEMPLO DE USO

### DEFINICIÓN DE LA RED

En la carpeta TestData encontrará un set de datos con el cual puede hacer una prueba del funcionamiento del programa, en este caso se incluye la información de 2 grafos (archivo dataNetSmall.txt y dataNetVerySmall.txt).

Para representar la siguiente red usaremos el archivo Para este ejemplo se hará uso del archivo dataNetVerySmall.txt.

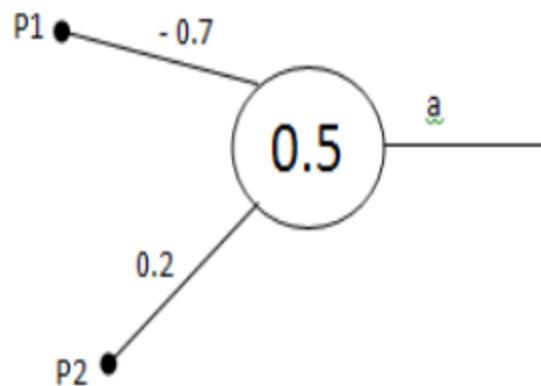


Figura 1. Red de prueba

Como puede ver en la figura anterior esta red tiene 2 entradas, 0 capas intermedias y 1 salida, esta es su topología. Por lo tanto la primera línea que corresponde a la topología debe quedar como la siguiente:

```
1 2 1
```

A continuación es necesario definir los valores de umbra que tienen los nodos de la red, en este caso existe solo uno nodo por lo tanto en la línea 2 de el archivo que describe la red está la siguiente información:

```
2 0.5
```

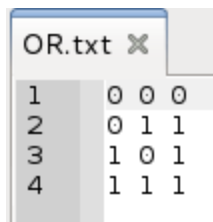
Por último se hace necesario definir los pesos de las conexiones o aristas entre los nodos, tenga en cuenta que el tamaño total en nodos de la red es 3, ya que cada entrada cuenta como un nodo adicional, por lo tanto el tamaño de la matriz de pesos para este ejemplo es de 3x3, los nodos que no tienen conexiones entre sí se les debe asignar un valor de 0.0

```
3 0.0 0.0 -0.7
4 0.0 0.0 0.2
5 -0.7 0.2 0.0
```

Cómo puede ver en las líneas anteriores la línea 3 equivale a las conexiones de la entrada 1 (p1) con los demás nodos y así sucesivamente, por lo tanto la conexión entre la entrada 1 (p1) y el nodo de salida es 0.7. Lo mismo para los demás conexiones.

## DATOS DE PRUEBA

En la carpeta TestData también se incluyen algunos Dataset no muy complejos para hacer prueba del entrenamiento de la red, entre ellos están las compuertas lógicas AND y OR (archivos AND.txt y OR.txt). Por cada línea de estos archivos hay un caso de prueba para la red, como esta red tiene dos entradas y una salida el archivo tendrá 3 columnas, la primera y la segunda equivalen a las entradas y la tercera equivale a la salida esperada.



```
OR.txt x
1 0 0 0
2 0 1 1
3 1 0 1
4 1 1 1
```

## EJECUCIÓN DEL PROGRAMA

A continuación se muestra una imagen de la ejecución del programa.

```

null3d@igun:~/workspace/RedNeuronal$ ./RedNeuronal
*****MENU PRINCIPAL*****
Bienvenido a un simulacion de una red nuroal por medio de grafos
A) Cargar datos desde archivos
B) Cargar datos desde linea de comandos
C) Empezar iteraciones
D) Usar datos por defecto
S) Salir
Ingrese opción deseada:A

Introduzca nombre de archivo que contiene la descripcion de la red:TestData/dataNetVerySmall.txt
Introduzca nombre de archivo de entrenamiento:TestData/OR.txt

*****MENU PRINCIPAL*****
Bienvenido a un simulacion de una red nuroal por medio de grafos
A) Cargar datos desde archivos
B) Cargar datos desde linea de comandos
C) Empezar iteraciones
D) Usar datos por defecto
S) Salir
Ingrese opción deseada:C

```

Como puede notar la primera opción seleccionada fue “Cargar datos desde archivo”, posteriormente el programa solicita los nombres de los archivos de entrenamiento y que describen la estructura de la red. Por último el programa vuelve al menú principal y mediante la opción C se le indica al programa que empiece la simulación.

## INTERPRETACIÓN DE LA SALIDA

### TÉRMINOS CLAVE

#### ITERACIÓN

Propaga la entrada por toda la red, obtiene la salida y modifica la red para que se adecue a la salida.

#### RECORRIDO

Corresponde a el número de vez que se ha recorrido un archivo de entrenamiento completamente.

#### EDICION

Número de veces que se modificó la red para adaptarla a la salida

## DISPONIBILIDAD

En el siguiente repositorio de GitHub encontrará el código de fuente y los datos de prueba de esta herramienta. También se incluyen los binarios.

<https://github.com/alejo0317/RedNeuronal>

## LICENCIA



Implementacion de red neuronal mediante grafos por Cristian Rojas se encuentra bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0 Unported](#).  
Basada en una obra en <https://github.com/alejo0317/RedNeuronal>.