

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Análisis evolutivo de la igualdad de género y de su impacto en  
los informes de sostenibilidad GRI**

**Carlos Alberto Guerrero Rodríguez  
Tutor: David Renato Domínguez Carreta**

**Mayo 2016**



# **Análisis evolutivo de la igualdad de género y de su impacto en los informes de sostenibilidad GRI**

**AUTOR: Carlos Alberto Guerrero Rodríguez**

**TUTOR: David Renato Domínguez Carreta**

**Dpto. Ingeniería de Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Mayo de 2016**





# Resumen

Este Trabajo Fin de Grado tiene como objetivo analizar la posibilidad de usar un nuevo sistema de aprendizaje automático que de mejores resultados para los informes de sostenibilidad GRI (Global Reporting Initiative). Se le hará varias pruebas a dicha red para poder analizar su efectividad.

El informe de sostenibilidad GRI ha servido para que las empresas hagan informes sobre su sostenibilidad. Estos informes se centran en 3 puntos importantes: Economía, Sociedad y Medio Ambiente.

La capacidad de predecir los posibles resultados de los informes de sostenibilidad que presentan las empresas puede resultar muy conveniente para posibles inversores. Pueden demostrar, estas empresas, que son viables y apuestas seguras para que estos inversores para ganar dinero, y la empresa conseguir inversión para seguir mejorando.

También puede servir para la población en general, estos informes explican que avances han hecho las empresas para mejorar la sociedad en la que viven, como por ejemplo que avances en derechos humanos han hecho, como la igualdad de género de sus trabajadores, donaciones, tareas comunitarias... o como han contribuido en preservar el medio ambiente, importante si queremos sobrevivir en el futuro y evitar que debido a los cambios como la extinción de especies o la desertización traerían graves problemas que provocarían grandes cambios a nuestra sociedad para peor. Por tanto los informes nos ayudarían a conocer que avances han hecho para reducir la contaminación, su búsqueda en energías renovables o reducir su huella en la naturaleza. Estos informes podrían ayudar a los gobiernos decidir a qué empresas ayudar o premiar por sus esfuerzos en mejorar la sociedad.

El método de predicción que usaremos de referencias es la red neuronal binaria, es decir, usa 2 estados para las neuronas. Podemos predecir hasta 2007 con esta red.

En este trabajo nuestro objetivo es estudiar una alternativa de método de predicción usando una red neuronal ternaria, es decir, que las neuronas pueden tener 3 estados. Compararemos los resultados con los de la red neuronal binaria y haremos varios ajustes para ver si podemos mejorarla aún más. También probaremos cuan robusto es nuestra red desordenando los años y añadimos ruido a los patrones.

## Palabras clave

Red neuronal binaria, red neuronal ternaria, neurona, patrón, Sociedad, Medio Ambiente, GRI, ANN, Simulación, Sostenibilidad, Predicción.

# Abstract

This Bachelor Thesis has as objective analyze the possibility of using a new system of learning that give better results for the sustainability reports GRI(Global Reporting Initiative). We will make some tests to the network to analyze its effectiveness

The sustainability reports GRI have served to the business as they make reports about their sustainability. These reports are centered in 3 important points: Economy, Society and environment.

The capacity to predict possible new results of the sustainability reports that present business can be very convenient for possible investors. They can demonstrate, the business, that they are viable and safe bets for these investors to get money and the business to get money to continue improving.

Also they can serve to the population in general, these reports explain their contributions to improve the society where they live, for example their improvements in human rights like genre equality of their workers, donations community Works... or how they contributed in preserve the environment, it is very important if we want to survive in the future and avoid that due to the changes like the extinction of species and desertification bring big problems that would provoke great changes to our society for the worse. So these reports can help us to know what advances they did to reduce their contamination, their search to renewable energy or reduce their footprint in the nature. These reports can help the government to decide to what business help or reward for their efforts in improving the society.

The method of prediction that we will use of reference is the binary neural network, it uses 2 states for the neurons. We can predict to 2007 with these network

In this thesis our objective is to study an alternative of prediction method using a ternary neural network, it uses 3 states for the neurons. We will compare the results with the binary neural network and we will make some adjustments to see if we can improve it further. Also we will try how strong it is our neural network against disordering years and adding noise to the patterns.

## Keywords

binary neural network, ternary neural network, neuron, pattern, Society, Environment, GRI, ANN, Simulation, sustainability, Prediction.





## *Agradecimientos*

*Quiero dar las gracias a mi familia por su apoyo durante toda mi vida.*

*También a mis amigos, Borja, Dani, Álvaro, Andrés, Edu y Alex que me ayudaron en las prácticas y acompañarme todos los años de la universidad.*

*Los profesores que he tenido a lo largo de la carrera por lo que me enseñaron y resolver mis dudas siempre que podían.*



## INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Organización de la memoria.....	2
2	Estado del arte .....	5
2.1	Redes Neuronales .....	5
2.1.1	Introducción.....	5
2.1.2	Ventajas .....	5
2.1.3	Aplicaciones .....	6
2.1.4	Red ternaria y red binaria .....	6
2.2	GRI .....	7
2.2.1	Introducción.....	7
2.2.2	Aspectos .....	8
2.2.3	Economía.....	8
2.2.4	Medio ambiente .....	8
2.2.5	Social .....	8
2.2.6	Relación con la red neuronal .....	9
2.3	Economía.....	9
2.3.1	Introducción.....	9
2.3.2	Relación con la sostenibilidad .....	9
3	Diseño.....	11
3.1	Diseño del programa.....	11
3.1.1	Lectura de los datos de GRI .....	11
3.1.2	Creación de la red neuronal ternaria.....	12
3.1.3	Clase para dibujar las gráficas .....	15
4	Desarrollo .....	17
4.1	Introducción.....	17
4.2	Desarrollo de la red neuronal ternaria .....	17
4.3	Posibles mejoras .....	22
4.3.1	<i>Usa una constante C</i> .....	22
4.3.2	<i>Usar una matriz de patrones con pesos</i> .....	22
4.4	Estudio de la robustez de la red neuronal .....	23
4.4.1	<i>Capacidad de aprender de años desordenados</i> .....	23
4.4.2	<i>Ruido capaz de soportar</i> .....	23
5	Integración, pruebas y resultados .....	25
5.1	Introducción.....	25
5.2	Gráfica normal.....	25
5.3	Mejoras .....	27
5.3.1	<i>Usa una constante C</i> .....	27
5.3.2	<i>Usar una matriz de patrones con pesos</i> .....	29
5.4	Robustez .....	30
5.4.1	<i>Capacidad de aprender de años desordenados</i> .....	30
5.4.2	<i>Ruido capaz de soportar</i> .....	31
6	Conclusiones y trabajo futuro.....	32
6.1	Conclusiones.....	32
6.2	Trabajo futuro .....	32
	Referencias .....	35

Glosario .....	37
Anexos.....	I
A      Manual de instalación.....	I
B      Manual del programador .....	III
C      Anexo ... ..	- 1 -

## INDICE DE FIGURAS

FIGURA 2-1: FORMULA DE ACTIVACIÓN DE LA RED BINARIA .....	6
FIGURA 2-2: FORMULA DE ACTIVACIÓN DE LA RED TERNARIA COMPARADA CON LA BINARIA.....	7
FIGURA 3-1: FORMULA PARA MOSTRAR LOS PATRONES PARA LA RED NEURONAL .....	11
FIGURA 3-2: EJEMPLO DE TABLA EXCEL QUE SE USA EN EL PROGRAMA PARA LOS DATOS DE LAS EMPRESAS DESDE EL 1999 HASTA 2013.....	12
FIGURA 3-3: VALORES DE M QUE DEMUESTRAN LOS PATRONES RECUPERADOS CON LA RED NEURONAL BINARIA. USA LOS DATOS DE N = 5897, K = 200 Y T=100. A LA IZQUIERDA ESTÁ EL RESULTADO CON EL MÉTODO ONLINE. A LA DERECHA ESTÁ EL RESULTADO CON EL MÉTODO OFFLINE.....	13
FIGURA 3-4: EJEMPLO DE LOS DISTINTOS EFECTOS QUE TIENEN LOS VALORES M Y L SOBRE LAS NEURONAS .....	14
FIGURA 3-5: DIAGRAMA DE FLUJO DEL PROGRAMA. ....	15
FIGURA 3-6: EJEMPLO DE LA LIBRERÍA FREECHART USANDO UNA GRÁFICA DE TARTA. ....	16
FIGURA 3-7: EJEMPLO DE LA LIBRERÍA FREECHART USANDO UNA GRÁFICA. ....	16
FIGURA 4-1: REPRESENTACIÓN DE LOS PATRONES .....	17
FIGURA 4-2: REPRESENTACIÓN DE LAS NEURONAS .....	17
FIGURA 4-3: FÓRMULAS DE M, Q Y N.....	18
FIGURA 4-4: FÓRMULA DE ACTIVIDAD.....	19
FIGURA 4-5: FÓRMULA DE L Y ETA .....	19
FIGURA 4-6: FÓRMULA DE I .....	19
FIGURA 4-7: FÓRMULA DE IP.....	19
FIGURA 4-8: FÓRMULA DE H, J Y K.....	20
FIGURA 4-9: FÓRMULA DE ACTIVACIÓN DE LA NEURONA, ACTIVADOR, POTENCIAL Y ESCALÓN ...	21

FIGURA 4-10: FORMULAS DE J Y K CON UNA CONSTANTE C .....	22
FIGURA 4-11: FUNCIONES PARA DAR PESOS A LOS AÑOS, I ES EL NUMERO DE INDICADORES DE UN AÑOS, N EL NÚMERO DE EMPRESAS.....	23
FIGURA 5-1: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CONSTANTE DE 0,5 .....	25
FIGURA 5-2: VALORES DE M QUE DEMUESTRAN LOS PATRONES RECUPERADOS CON LA RED NEURONAL BINARIA. USA LOS DATOS DE N = 5897, K = 200 Y T=100. A LA IZQUIERDA ESTÁ EL RESULTADO CON EL MÉTODO ONLINE. A LA DERECHA ESTÁ EL RESULTADO CON EL MÉTODO OFFLINE.....	26
FIGURA 5-3: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CON VALOR C DE 0 .....	27
FIGURA 5-4: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CON UN VALOR C DE 1.....	28
FIGURA 5-5: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CONSTANTE C DE 0,5.....	29
FIGURA 5-6: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CONSTANTE C DE 0,5 CON LOS AÑOS DESORDENADOS .....	30
FIGURA 5-7: GRÁFICA CON LOS RESULTADOS DEL ALGORITMO CON UNA MUESTRA DE 5897 EMPRESAS, EN UN LAPSO DE 15 AÑOS Y CONSTANTE C DE 0,5 CON UN RUIDO DEL 0,1.....	31

# 1 Introducción

---

## 1.1 Motivación

Durante los últimos años ha habido una creciente importancia en las empresas en conseguir ser más sostenibles. Esta importancia viene debido a la concienciación de la población sobre temas del medio ambiente, sociales y económicos.

Los ciudadanos de los distintos países quieren mejoras en sus vidas y que las empresas se presten a satisfacerlas. En la sociedad por ejemplo quiere mejoras para que podamos ir a un mundo más ecológico y respetuoso con la naturaleza para que nuestros descendientes puedan respirar tranquilos libres de la polución, mejoras económicas

Uno de los métodos para que las empresas comuniquen sus intereses en mejorar las vidas de los habitantes de las ciudades donde trabajan estas empresas es el informe de Sostenibilidad GRI.

En este informe las empresas deben explicar qué están haciendo para mejorar o permanecer en el mercado, que están haciendo para mejorar sus condiciones, como progresan a una empresa más sostenible en los 3 temas que estudia GRI, medio ambiente, economía y la sociedad.

Para poder predecir el comportamiento proponemos una métrica ANN para predecir los resultados de los informes de sostenibilidad de las empresas. Poder predecir los comportamientos en las empresas ha demostrado ser de gran ayuda para la economía y podemos aprovechar la potencia de los ordenadores para reconocer patrones de conducta en las empresas y predecir sus futuros resultados. Puede ser importante tanto para los inversores, al poder conocer mejor el riesgo de invertir en una compañía como para los clientes que pueden saber mejor que empresa de su entorno se esfuerza más por conseguir una mayor sostenibilidad.

La ANN que tenemos como referencia es la red neuronal binaria, en esta red todos los nodos están conectados con todos. Usa una métrica Hopfield, muy útil cuando los conjuntos de datos están ordenados temporalmente, en este caso los patrones de las empresas están ordenados desde 1999 hasta 2013 divididos en años.

Es en este informe donde nos centraremos en mejorar el rendimiento de la red neuronal binaria que se usa en el artículo “Modeling sustainability report scoring sequences using an attractor network” con una red neuronal ternaria. La razón de esto es poder saber si al añadirle a los valores de los patrones y las neuronas un valor más conseguimos que la red neuronal sea más precisa y por tanto conseguir mejores resultados en los valores que usaremos para reconocer si tiene éxito en reconocer patrones con las neuronas.

## **1.2 Objetivos**

El informe de sostenibilidad GRI es uno de los informes de sostenibilidad que se pueden considerar estándar para las empresas. Cada vez son más empresas que usan las guías para realizar informes de GRI al mostrarse como guía útil y eficaz para conseguir informes de gran calidad y poder por tanto seguir mejor el progreso de la empresa tanto en el mercado como en conseguir que sea más sostenible.

Por tanto conseguir una métrica que sirva para predecir los resultados de los informes de sostenibilidad es muy importante por su cantidad de posibilidades para la economía y la sociedad en general conocer el progreso de las empresas estudiadas por la red. Actualmente disponemos de un modelo que ha demostrado su eficacia para conseguir predecir los resultados de los informes de sostenibilidad tras aprender de años anteriores usando una red neuronal binaria.

En este proyecto nuestro principal objetivo es estudiar, y si es posible demostrar, que podemos tener una alternativa mejor que una red binaria, en este caso estudiaremos las posibilidades de la red neuronal ternaria como posible alternativa más eficaz para predecir resultados en los informes de sostenibilidad.

## **1.3 Organización de la memoria**

La memoria consta de los siguientes capítulos:

- Primer capítulo introducción al TFG, en este apartado hablaremos brevemente de los objetivos marcados en este proyecto.
- Segundo capítulo, hablaremos del estado del arte, aquí explicaremos las redes neuronales, el informe de sostenibilidad GRI y los apartados que la componen, la razón por la que se relacionan este informe con las redes neuronales.
- Tercer capítulo, hablaremos del diseño usado para poder saber si la red neuronal ternaria es mejor que red neuronal binaria para el informe de sostenibilidad GRI y que vamos a usar para representar las gráficas y como leeremos la tabla con las empresas que han presentado un informe de sostenibilidad
- Cuarto capítulo, explicaremos como hemos desarrollado el algoritmo para hacer la red neuronal y especificaremos como leer la tabla y como mostramos la gráfica con los resultados obtenidos.
- Quinto capítulo, explicaremos que resultados hemos obtenido de las pruebas hechas, mostraremos las gráficas obtenidas en varias situaciones, usando una constante para

modificar el resultado, la tolerancia al ruido, las compararemos con la red neuronal binaria.

- Sexto y último capítulo, mostramos las conclusiones obtenidas con este proyecto y hablaremos de posibles proyectos que podemos realizar en el futuro si la red neuronal ternaria se muestra como mejor alternativa que la red neuronal binaria.





## 2 Estado del arte

---

### 2.1 Redes Neuronales

#### 2.1.1 Introducción

La red neuronal es uno de los métodos de aprendizaje automático, inspirado en las neuronas del cerebro. Está formada por:

- Neuronas que reciben señales que activan las neuronas.
- Patrones que usan las neuronas para aprender y predecir nuevos patrones.
- Conexiones entre las neuronas, se establece que neuronas están conectadas con otras neuronas.

Para poder modificar las neuronas se usan 3 funciones:

- Una función de activación: cuando pasa un umbral la neurona se considera activada o desactivada.
- Una función de propagación: que suma todas las entradas de la neurona para definir si es la conexión es positiva o negativa.
- Una función de transferencia: Usando la función de activación, sirve para acotar la salida de la neurona.

La idea de usar un sistema de aprendizaje automático basado en el cerebro humano, viene de 1943 con los primeros modelos de redes neuronales. Se pueden usar dos métodos para conseguir la red neuronal, el primero es hacer un modelo matemático y a partir de ahí hacer un algoritmo que pueda resolver el problema, y el segundo usando una base de datos, leer los datos para que la red neuronal aprenda las propiedades deseadas, en este método en vez de modelar matemáticamente se diseña la red mediante la selección de variables, procesar la información entrante que se usara para el entrenamiento y seleccionar un modelo de red.

#### 2.1.2 Ventajas

- **Aprendizaje**
- **Auto organización**
- **Tolerancia a fallos**
- **Flexibilidad**
- **Tiempo real**

### 2.1.3 Aplicaciones

Las redes neuronales sirven para aplicaciones donde no se puede reconocer un modelo de aprendizaje automático identificable en un principio, pero se conoce la base de datos para entrenar dicho modelo.

Son también ideales por ser robustas ante el ruido y poder funcionar en paralelo.

Buenos ejemplos de aplicaciones son el reconocimiento de voz, de huellas, predicciones del tiempo, del mercado o de identificación de fraude.

La red Kohonen es un ejemplo de red aplicada a problemas matemáticos con una complejidad razonable como el problema del viajante. Usa un algoritmo de vecindad para preservar las propiedades topológicas de los datos de entrada. Para aprender busca las partes del mapa que responde de forma similar a un grupo de patrones de entrada.

Se puede usar junto con los algoritmos genéticos para la creación de robots. Aquí el genoma usado en los algoritmos genéticos es creado mediante parámetros de la red neuronal y la red capta el comportamiento del robot controlado en el momento.

### 2.1.4 Red ternaria y red binaria

La red binaria usual usa como estados 0 y 1, emulando los estados de la neurona activado y desactivado.

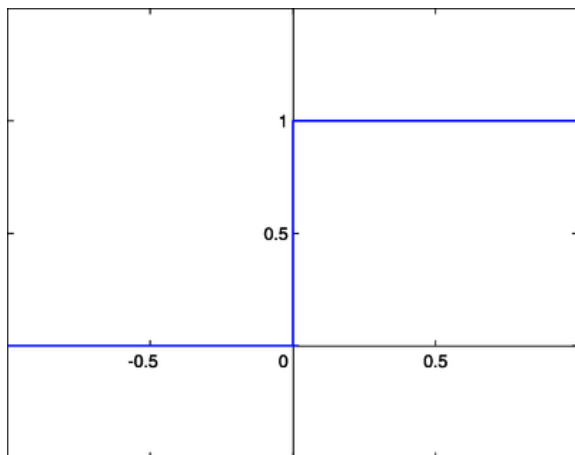
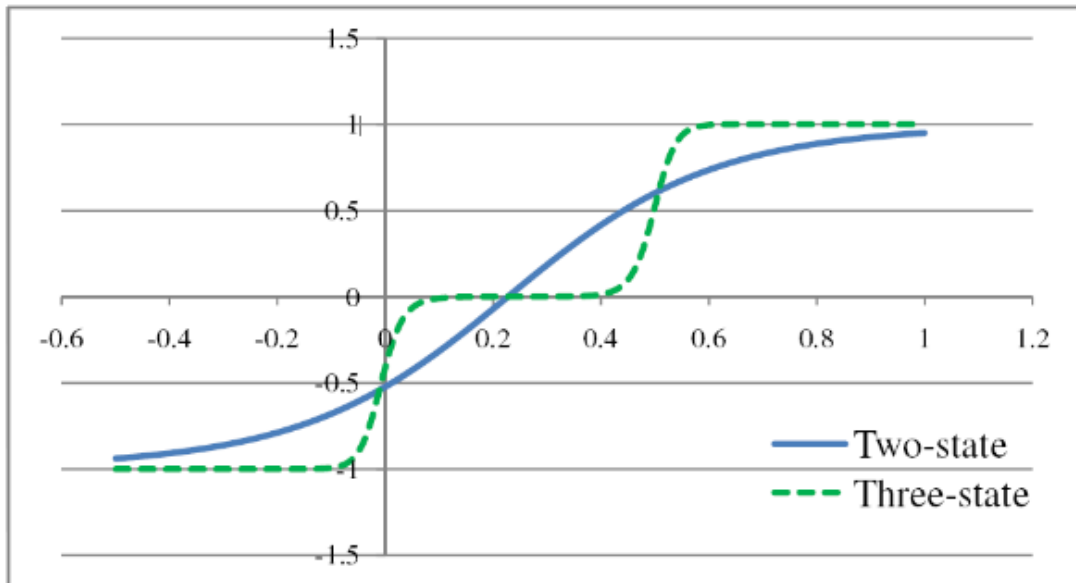


Figura 2-1: Formula de activación de la red binaria

Mientras la ternaria usa 3 estados -1, 0 y 1, este aumento de estados permite una mayor precisión para las redes.



**Figura 2-2: Fórmula de activación de la red ternaria comparada con la binaria**

La diferencia en usar más estados para las neuronas, permite un aumento de la precisión en los casos de predicción de comportamiento, al tener un entrenamiento más exacto. La red ternaria también ha sido mostrada más eficaz si usamos un ANN que tenga varios estadios, donde los resultados se muestran mucho mejores que la versión convencional.

## **2.2 GRI**

### **2.2.1 Introducción**

La **Iniciativa de Reporte Global** o *Global Reporting Initiative*, es la primera iniciativa para crear memorias de sostenibilidad para que las empresas reporten sus avances por la sostenibilidad ambiental, económica y social. Promovido por las Naciones Unidas para crear informes sobre la sostenibilidad con la misma calidad que se crean informes financieros.

Es una guía que permitirá a las empresas crear informes de sostenibilidad fácilmente siguiendo las guías que proporcionan.

La utilidad de estas guías permitirá conocer el progreso de las empresas con el desarrollo de un plan sostenible y por tanto un método de motivación para mejorar de dichas empresas al mostrarse tanto más viables económicamente como una mejor opinión pública al estar más comprometido con la sociedad y el medio ambiente.

### **2.2.2 Aspectos**

La memoria GRI se centra en los aspectos materiales de los que trata, económicos, medio ambientales, sociales, como impactos de la empresa. El informe contempla dos categorías de aspectos:

Aspectos generales: estos aspectos ponen en contexto tanto la empresa, como el informe, describiendo como se ha llevado a cabo el informe, la situación de la empresa, etc. Esta parte del informe es común a todas las empresas independientemente de su situación. Se divide a su vez en 7 categorías, algunas de ellas son, perspectiva estratégica para abordar los asuntos de sostenibilidad, relación con los grupos de interés, el gobierno, la sociedad en general y la ética.

Aspectos específicos: Estos aspectos explican cómo funcionan sus impactos en la economía, medio ambiente y sociedad y su relación con la sostenibilidad. Por tanto describen las causas de sus impactos, como los desarrollan y sus consecuencias en la sostenibilidad. También presenta los indicadores, que sirven para explicar la situación de la empresa facilitando información mediante unos indicadores, mayormente cuantitativos, que se pueden comparar frente a otras empresas y además facilitar el seguimiento de su progreso por la sostenibilidad.

También ofrece una guía para tratar puntos específicos a un sector determinado, como el impacto del ruido de un aeropuerto.

### **2.2.3 Economía**

Esta sección recoge los datos financieros y el impacto en la economía en las regiones en las que trabaja. Para hacerlo sigue las indicaciones de cuentas auditadas usando las Normas Internacionales de Información Financiera, las interpretaciones de las normas del Consejo Internacional de Normas Contables y las reglas regionales para la elaboración de informes financieros.

### **2.2.4 Medio ambiente**

Aquí se recogen los datos del impacto medio ambiental de la organización. Estudia el impacto de los gastos, como energía y agua, y los productos, emisiones, efluentes o desechos. También investiga el impacto que causan los servicios y productos en los ecosistemas donde trabajan.

### **2.2.5 Social**

En esta categoría estudia las Prácticas laborales, que se cumpla el trabajo digno, los derechos humanos, su impacto en la sociedad y la responsabilidad en el trabajo de sus productos y servicios.

## **2.2.6 Relación con la red neuronal**

Las redes neuronales pueden ser de ayuda para predecir futuros comportamientos en las empresas que presenten informes de sostenibilidad. El mayor beneficio de estas predicciones será para inversores y la sociedad al poder prever la evolución de las empresas en los distintos campos que ocupa GRI.

## **2.3 Economía**

### **2.3.1 Introducción**

La economía es el área de producción, distribución, comercio y consumo de bienes y servicios. Se puede definir como una ciencia social debido al gran impacto que puede tener en la sociedad actual y estudiar el comportamiento humano para crear modelos matemáticos.

Estudia también la utilización de los recursos limitados para conseguir su valor óptimo para poder ser utilizados de la mejor manera. También hace el estudio de cómo se crean recursos y se destruyen.

La economía es uno de los puntos más importantes que debe tener cualquier empresa para poder funcionar correctamente y conseguir beneficios.

La economía es uno de los puntos donde las ideologías más hacen hincapié para poder manejarla a voluntad. Así pueden conseguir el control de los recursos y poder explotarlo como vean conveniente.

Es una ciencia además que se adapta y apropia a otras ciencias, por ejemplo para poder predecir el comportamiento de los mercados debe poder usar tanto las matemáticas como ciencias sociales como la sociología para poder predecir el comportamiento humano y predecir el impacto que tienen las acciones que hagamos o la medición del valor de los productos para poder repartir mejor el dinero y obtener el mayor beneficio posible.

### **2.3.2 Relación con la sostenibilidad**

Tras la última crisis se ha hecho más evidente la necesidad de un plan de sostenibilidad más efectivo para poder tanto seguir en el mercado como poder estimar el impacto que tiene tanto en la sociedad como en el medio ambiente.

Esto ha hecho que las empresas deban demostrar que son sostenibles para poder atraer inversores. Pero no solo también ha de buscar la innovación, la concienciación y la promoción de un consumo sostenible de los productos, debe por tanto asegurarse que el consumo de productos en el mercado pueda estar equilibrado con los ciclos de renovación de sus materias primas.

Pero no solo se debe asegurar que el consumo se mantiene a nivelado con los ciclos de renovación de las materias primas. También debe buscar incentivar el consumo de energías limpias para reducir la contaminación y aumentar la investigación en métodos de producción que permitan ser más eficientes, pudiendo así producir más con menos materias primas y por tanto aumentar los productos de consumo sin que el ecosistema se resienta.

## 3 Diseño

---

### 3.1 Diseño del programa

#### 3.1.1 Lectura de los datos de GRI

En esta sección se hablara del método para leer los datos de las empresas de los informes de GRI.

Para poder leer los datos he decidido que se hará usando un archivo Excel en formato xlsx. De esta forma será sencillo leer los datos de las empresas en todos los años, ya que hay celdas donde no hay nada escrito y será fácil leer esas ausencias con el valor más bajo de las redes neuronales, -1 en el sistema ternario y decidir si usar 0 o 1 según el número de indicadores que tenga la celda. A diferencia del otro tipo de archivo, el binario, donde podemos confundir fácilmente los años si hay alguno en blanco y tener errores como se ha leído el dato de un año como si fuese el del año anterior y dando por lo tanto errores como que en vez de leer 15 años hemos leído solo 13.

$$F(x) = \begin{cases} -1 & x = 0 \\ 0 & x > 0 \ \& \ x < 3 \\ 1 & x > 3 \end{cases}$$

Figura 3-1: Formula para mostrar los patrones para la red neuronal

La lectura de la tabla se hará con la librería POI, usando las funciones para xlsx.

La clase que lee el informe de GRI creara una matriz donde los patrones ya están listos para usarlos en la red neuronal ternaria.

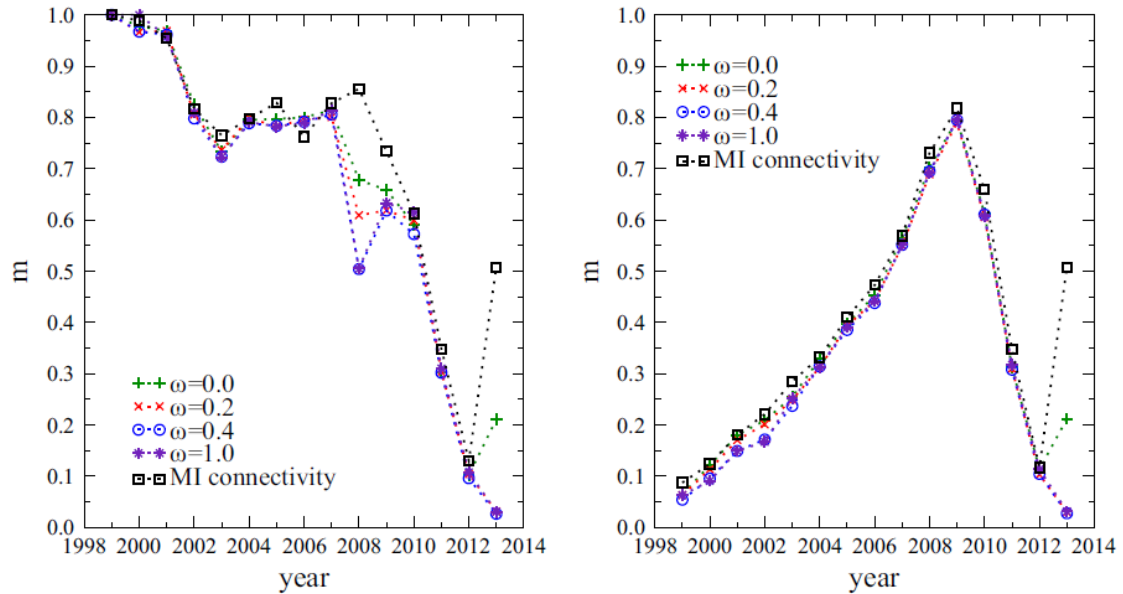


2004	2005	2006	2007	2008
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
G2, IA, Sd	G2, IA, Sd	G2, IA, Sd	G3, A+, 3p	G3, A+, 3p
0	0	0	0	G3, U
0	0	0	0	G3, U
0	0	0	0	0
0	0	0	0	G3, C, Sd
0	0	0	G3, C, Sd	G3, C, Sd
0	0	0	0	G3, C, Sd

**Figura 3-2: Ejemplo de tabla Excel que se usa en el programa para los datos de las empresas desde el 1999 hasta 2013.**

### 3.1.2 Creación de la red neuronal ternaria

En esta sección se hablara de cómo se realizó la clase de la red neuronal ternaria. Para realizar esta red neuronal se ha tomado como base el trabajo de la red neuronal binaria que se hizo para el informe GRI. Dicho sistema está basado crea a partir de un archivo Excel sobre las empresas que reportaban sus informes GRI durante varios años, 1999-2013. Los patrones y las neuronas se usaban solo los valores 0 y 1, aquí en la red neuronal ternaria usaremos los valores -1, 0 y 1. Nuestro objetivo es mejorar los resultados de la red neuronal binaria con la red neuronal ternaria, en la binaria el resultado son la recuperación de patrones desde 1999 hasta 2007, para mejorarlo deben recuperarse más años.

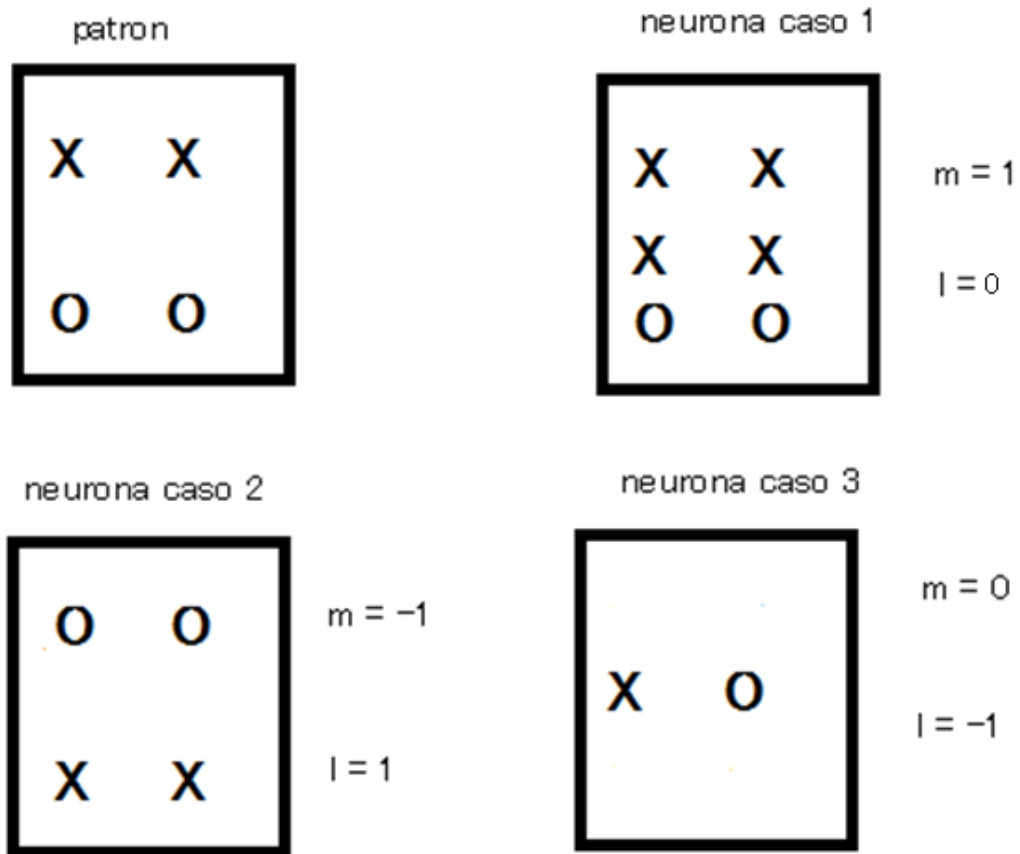


**Figura 3-3: Valores de  $m$  que demuestran los patrones recuperados con la red neuronal binaria. Usa los datos de  $N = 5897$ ,  $K = 200$  y  $t=100$ . A la izquierda está el resultado con el método online. A la derecha está el resultado con el método offline.**

Para realizar la red neuronal ternaria usaremos las fórmulas del artículo “Three-state Neural Network: from Mutual Information to the Hamiltonian” En este artículo se detalla la utilidad de las fórmulas que usaremos para este trabajo y que explicaremos en detalle en la siguiente sección.

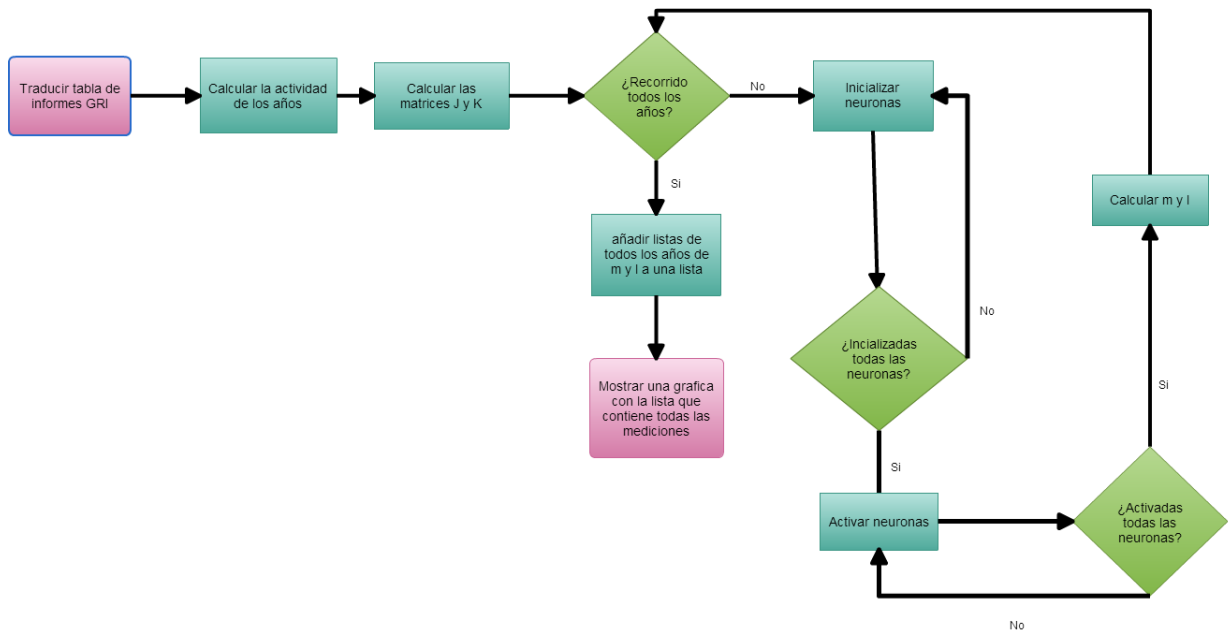
En las pruebas nos centraremos en las medidas  $m$  y  $l$ . En función de sus valores podremos saber si, la neurona recuerda exactamente donde está el valor del patrón, si añade un ruido aparte de adivinar el valor, lo pone al revés o solo añade ruido.

Por lo tanto lo que nos interesa de conseguir estos valores es que estén en lo más alto posible. Si lo están sabremos no solo el valor del patrón sino su posición en el sistema.



**Figura 3-4: Ejemplo de los distintos efectos que tienen los valores  $m$  y  $l$  sobre las neuronas**

La forma de leer la tabla Excel es la de contar los indicadores que tienen en cada celda, no se contempla en este proyecto que pueda haber distintos pesos según el indicador.



Original: [http://en.wikipedia.org/wiki/File:\(1\)\\_2008-04-07\\_Information\\_Management\\_Help\\_Desk.jpg](http://en.wikipedia.org/wiki/File:(1)_2008-04-07_Information_Management_Help_Desk.jpg)

**Figura 3-5: Diagrama de flujo del programa.**

Como vemos en el diagrama de flujo el programa va a tener 3 bucles anidados como parte principal del programa. El diagrama de flujo mostrado es el diagrama del programa principal. Después de conseguir los patrones y conocer la actividad de cada año procedemos a crear la red neuronal ternaria empezando con las matrices J y K que no necesitan estar dentro del bucle porque solo usan la matriz de patrones. Cuando se empiezan a recorrer los años primero se deben inicializar todas las neuronas añadiéndoles el valor del patrón correspondiente, después usamos la función de activación para activar todas las neuronas, este bucle además de recorrer todas las neuronas repite la función para cada neuronas hasta 10 veces para asegurarse que el valor es el correcto y que tras cada repetición no hay cambios en los valores. Tras activar las neuronas se empieza medir cuantas neuronas han aprendido correctamente usando las medidas de m y l. Finalmente después de recorrer todos los años añadimos las medidas de m y l a una lista que usaremos para hacer la gráfica que represente la evolución de m y l a través de los años.

### 3.1.3 Clase para dibujar las gráficas

En esta sección se hablara del método para dibujar las gráficas.

Para dibujar las gráficas he usado una librería de Java, llamada freechart, que permite dibujar varios tipos de gráficas y poder juntarlos en una misma gráfica. Dicha librería permite crearlas usando un código sencillo donde es fácil conseguir poner los datos en una lista para su posterior lectura en la gráfica y personalizarla con distintos tipos de gráficas. Para visualizarla he usado una interfaz sencilla donde solo necesite mostrar la imagen con la gráfica ya hecha.

Las gráficas que mostraremos serán las de las medidas de los valores de m y l que tienen cada año.

Algunos ejemplos de la librería de freechart son:

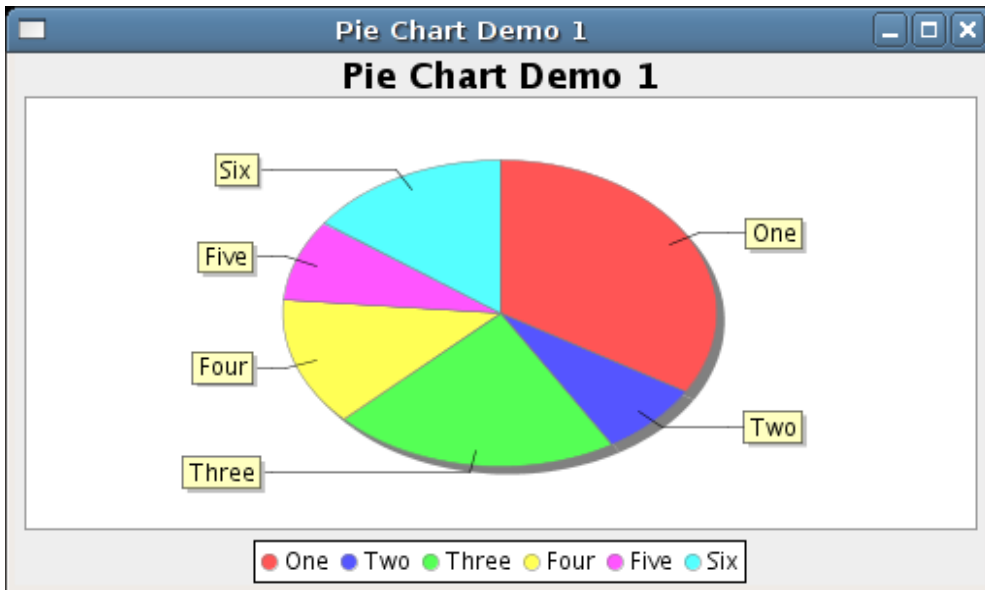


Figura 3-6: Ejemplo de la librería freechart usando una gráfica de tarta.

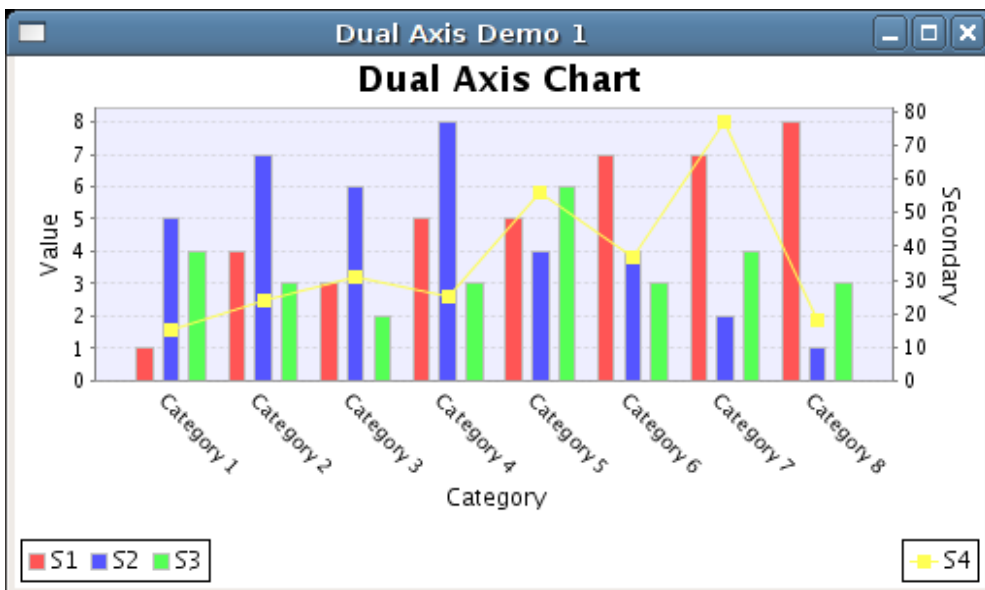


Figura 3-7: Ejemplo de la librería freechart usando una gráfica.

## 4 Desarrollo

---

### 4.1 Introducción

Para el desarrollo de las pruebas he puesto de un total de 5987 empresas, he preparado distintas pruebas para números de empresa desde 1000 hasta 5000 y cogiendo todos los años. De los resultados de las formulas se estudiarán  $m$ ,  $n$ ,  $l$  y  $I$  que son las medidas que corroboran el éxito de la red neuronal. Una vez hechas las pruebas se usan los resultados en una gráfica para ver mejor cuantos patrones son recuperados.

Para realizar estas pruebas se ha empleado un ordenador con una capacidad de 16 GB de RAM para poder guardar la red neuronal se hagan pruebas con números grandes de empresas, por ejemplo 5000, sin problemas.

### 4.2 Desarrollo de la red neuronal ternaria

Para desarrollar la red neuronal ternaria se ha usado las formulas del documento “Three-state Neural Network: from Mutual Information to the Hamiltonian” donde podemos leer que primero:

Las neuronas tienen 3 estados, por tanto los patrones también tienen 3 estados y son una matriz donde en las filas son las empresas y las columnas los años.

Lo que usaremos para medir el éxito de las neuronas son las formulas del solapamiento del patrón con el estado de la neurona,

$$\xi_i^\mu \in \{0, \pm 1\}, \mu = 1, \dots, p$$

**Figura 4-1: Representación de los patrones**

Los patrones por tanto seguirían la misma fórmula usando los 3 estados para representar el estado del patrón en unos años. Para poder decidir cómo asignar los 3 valores, usaremos -1 para cuando no hay valores y diferenciaremos 0 y 1 cuando el número total de indicadores de un patrón en un año supere un valor adecuado, en este caso el valor es 3.

$$\sigma_i \in \{0, \pm 1\}, i = 1, \dots, N.$$

**Figura 4-2: Representación de las neuronas**

La solapamiento de la neurona, m, el actividad neuronal, q, el solapamiento-actividad, n.

$$m_N^\mu \equiv \frac{1}{aN} \sum_i \xi_i^\mu \sigma_{i \rightarrow m} = \left\langle \langle \sigma \rangle_{\sigma|\xi} \frac{\xi}{a} \right\rangle_\xi,$$

$$q_{Nt} \equiv \frac{1}{N} \sum_i |\sigma_{it}|^2 \rightarrow q = \langle \langle \sigma^2 \rangle_{\sigma|\xi} \rangle_\xi,$$

$$n_{Nt}^\mu \equiv \frac{1}{aN} \sum_i |\sigma_{it}|^2 |\xi_i^\mu|^2 \rightarrow n = \left\langle \langle \sigma^2 \rangle_{\sigma|\xi} \frac{\xi^2}{a} \right\rangle_\xi.$$

**Figura 4-3: Fórmulas de m, q y n**

m, representa las recuperaciones del solapamiento, es un parámetro normalizado cuyos valores están dentro de [-1,1], que obtienen cuando en un año una neurona tiene un valor entre [-1,1]. Este valor comprende si el resultado es el correcto, sumara 1 o dará -1 si es 0 significa que no recupera información, por tanto nuestros mejores resultados serán siempre 1 o -1, preferiblemente 1 que significara que el resultado es siempre el correcto, -1 será el resultado si la neurona vale 1, pero el patrón es -1, es decir la neurona y el patrón tienen los signos cambiados.

q, representa la actividad neuronal de las neuronas, es una media de todas las neuronas y calcula cuantas están activadas, -1 o 1 y por tanto su valor está en un valor entre [0,1].

n, este parámetro no parece de mucha importancia para la evolución de la red, pero será un elemento importante de la información mutua, además de conocer el rendimiento de la red. Con n conoceremos cuantas neuronas coinciden con un patrón en activo, es decir si la neurona y el patrón coinciden en que son distintos de 0 sin importar el signo de los valores obtenidos.

La actividad de cada año la calculamos sumando todos los patrones que no de 0 y hacer la media total. Como cada año tiene distintos patrones activados se debe hacer un array para todas las actividades de los años.

$$a^\mu \equiv \frac{1}{N} \sum_i |\xi_i^\mu|^2$$

**Figura 4-4: Fórmula de actividad**

Buscamos una función que sea simétrica a cualquier permutación de los patrones de un año, al no ser conocidos inicialmente durante la recuperación. Esto requiere que la recuperación inicial de cualquier de cualquier patrón sea débil, por ejemplo que  $m$  se aproxime a 0. Por lo general  $a$ ,  $q$  y la  $a$  variable neurona al cuadrado son independientes de patrón al cuadrado, así que  $n$  es semejante a  $q$ . Por lo que podemos los parámetros:

$$l^\mu \equiv \frac{n^\mu - q}{1 - a} = \langle \sigma^2 \eta^\mu \rangle \quad \eta^\mu \equiv \frac{(\xi^\mu)^2 - a}{a(1 - a)}$$

**Figura 4-5: Fórmula de l y eta**

Que también desaparece cuando las neuronas y patrones son independientes.

Podemos usar esto para conseguir la función de Mutual Information, donde conseguimos una función que expandimos de cuando  $m$  y  $l$  son iguales 0:

$$I^\mu \approx \frac{1}{2} \frac{a}{q} (m^\mu)^2 + \frac{1}{2} \frac{a(1 - a)}{q(1 - q)} (l^\mu)^2.$$

**Figura 4-6: Fórmula de I**

Para conseguir toda la información de la red solo necesitamos obtener la información de todas las  $I$  de los años que usamos para crear la red neuronal.

$$I_{pN} = N \sum_\mu I^\mu$$

**Figura 4-7: Fórmula de Ip**

Si queremos simplificar el modelo, como querer que la actividad del patrón es parecido a la actividad de la neurona. Esto nos permite simplificar los mecanismos del modelo que serían difíciles de computar en otro caso.  $I$  depende de  $m$  y  $l$ . Sustituyendo los



parámetros y usando las definiciones de las formulas obtenemos otra forma de obtener la información de la red. H1 es el termino cuadrático y H2 es el termino bicuadrático. Las interacciones están distribuidas aleatoriamente por J y K. El primero es el mismo que el usado en el modelo Hopfield, que se usó para hacer la red neuronal binaria. El segundo está asociado con la orden cuadripolar del sistema.

$$\mathcal{H} = -I_p N \equiv \mathcal{H}_1 + \mathcal{H}_2 ,$$

$$\mathcal{H}_1 = -\frac{1}{2} \sum_{ij} J_{ij} \sigma_i \sigma_j , \quad J_{ij} = \frac{1}{a^2 N} \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu$$

$$\mathcal{H}_2 = -\frac{1}{2} \sum_{i,j} K_{ij} \sigma_i^2 \sigma_j^2 , \quad K_{ij} = \frac{1}{N} \sum_{\mu=1}^p \eta_i^\mu \eta_j^\mu$$

**Figura 4-8: Fórmula de H, J y K**

Para activar una neurona haremos uso de este activador y como variables de la función se usa el potencial, h, y el potencial neuronal, O. Para la función escalón, G, hacemos una resta entre variables y si sale mayor que 0 ponemos el valor como 1. Para conocer el signo usamos únicamente h, si h es positiva entonces la neurona es positiva y negativa si h es negativa.

$$\sigma_i = \text{sgn}(h_i) G(h, \theta)$$

$$h_i \equiv \sum_j J_{ij} \sigma_j,$$

$$\theta_i \equiv \sum_j K_{ij} \sigma_j^2,$$

$$G(h, \theta) = \Theta(|h| + \theta)$$

$$\Theta(x) \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

**Figura 4-9: Fórmula de activación de la neurona, activador, potencial y escalón**

Para tratar correctamente la activación de las neuronas hemos tenido en cuenta que la activación de las neuronas usando la función una sola vez puede dar lugar a incongruencias. Por tanto y con semejanza a otras funciones con un problema parecido como el PageRank iteraremos varias veces la función de activación, la condición para iterar será iterar un número determinado de veces, también podría ser una comparación con la iteración anterior y comprobar que no hay o hay muy poca diferencia entre los dos, que llamaremos condición de convergencia donde se espera que al final la iteración tengamos el resultado real de la activación.

Nos centraremos en las variables  $m$  y  $l$  para comparar sus valores con los de la actual red neuronal del informe GRI. El objetivo es que consiga recuperar más años para poder demostrar que esta red neuronal es mejor.

El tiempo de ejecución del algoritmo está condicionado por los bucles que tiene que recorrer el programa.

El tamaño del algoritmo está definido por las matrices de  $J$ ,  $K$ , patrones, actividades y las neuronas.

### 4.3 Posibles mejoras

En esta sección se va a estudiar posibles mejoras que pueden servir para mejorar los resultados.

#### 4.3.1 Usa una constante C

Para conseguir una mayor ganancia de información probaremos una forma de mejorar los resultados añadiendo una constante C en las fórmulas de J y K, estas fórmulas son muy importantes para calcular los valores del activador de la neurona. No podemos adivinar con certeza cuál es el valor correcto que funcionaría como peso para darle más valor una de las matrices sobre la otra.

Probaremos por tanto distintos valores para C, que modificarán los valores de m y l esencialmente y sabremos con certeza con qué valor de C les afecta menos negativamente y más positivamente a ambos usando de referencia los valores en la gráfica de valores normal con una constante que valore igual tanto J como K.

$$J_{ij} = C \frac{1}{a^2 N} \sum_{\mu=1}^p \xi_i^{\mu} \xi_j^{\mu}$$
$$K_{ij} = (1 - C) \frac{1}{N} \sum_{\mu=1}^p \eta_i^{\mu} \eta_j^{\mu}$$

Figura 4-10: Formulas de J y K con una constante C

#### 4.3.2 Usar una matriz de patrones con pesos

Los primeros años están muy descompensados si los comparamos con los últimos años donde los primeros no tienen apenas actividad mientras en los últimos hay un mayor número de indicadores. Para solucionar eso y equilibrarlo vamos a probar a hacer un sesgo donde en vez de hacer que la traducción de la tabla Excel en patrones solo cambie a 0 o 1 si tiene un número de indicadores fijado, lo haremos adaptativo según el año de la tabla. De esta forma cuando consigamos todos los indicadores de un año hacemos la media y lo dividimos en 3 partes donde en cada una para cada patrón según el número de indicadores que tenga tendrá un entero determinado.

De esta manera podremos nivelar el peso que tiene cada año en la generación de la red neuronal. Así los primeros años podrán ser más relevantes que antes.

$$S = I/N$$

$$B = 2 * S/3$$

$$F(x) \begin{cases} -1 & x < B \\ 0 & x \geq B \ \& \ x < 2B \\ 1 & x \geq 2B \end{cases}$$

**Figura 4-11: Funciones para dar pesos a los años, I es el número de indicadores de un año, N el número de empresas.**

## **4.4 Estudio de la robustez de la red neuronal**

### **4.4.1 Capacidad de aprender de años desordenados**

Para demostrar la independencia de los patrones a lo largo de los años y usarlo como forma de comprobar la robustez de la red neuronal queremos probar que resultados obtendríamos si los años estuviésemos desordenados, así podríamos ver su comportamiento si para los resultados de los patrones en el año 2010, por ejemplo, los patrones a aprender fuesen los del año 2000.

### **4.4.2 Ruido capaz de soportar**

Para demostrar la robustez del sistema vamos a probar cuánto ruido puede soportar mientras pueda leer de forma correcta los patrones de cada año. Por tanto aunque se espera una disminución de los resultados de cada gráfica no se espera que bajen demasiado y que

como mucho solo reduzca en un año el número de años que han aprendido satisfactoriamente los patrones.

Para incluir el ruido que pondremos dividiremos el porcentaje del ruido que habrá en 3 partes, -1,0 y 1 cuyo porcentaje de probabilidad será la multiplicación del porcentaje del ruido por el número de repeticiones del valor dividido por el número de filas.

Si se da la probabilidad de ruido cuando toque un patrón este cambiara según como este dividido le ruido en ese año. Si no se da la probabilidad de ruido el patrón se mantendrá como esta.

# 5 Integración, pruebas y resultados

## 5.1 Introducción

En esta sección hablaremos de los resultados obtenidos en la red neuronal. Hablaremos de las gráficas que se han obtenido y las compararemos con las gráficas de red neuronal binaria para comprobar si el rendimiento de la nueva red neuronal ternaria es mejor que la actual.

Los datos usados para obtener las gráficas son de 5897 empresas durante 15 años. Para usar de gráfica de referencia usamos 10 repeticiones, con un valor de constante C de 0.5.

Empezaremos hablando de los resultados actuales de cada variable para terminar hablando de las distintas mejoras y las pruebas de la robustez del programa.

## 5.2 Gráfica normal

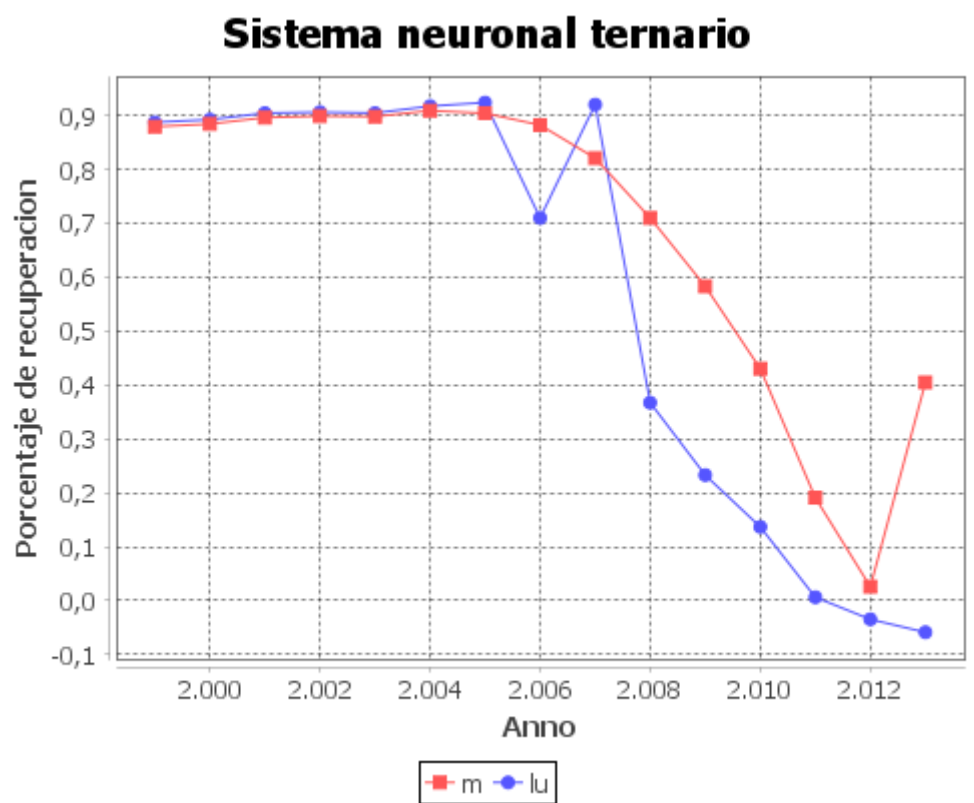
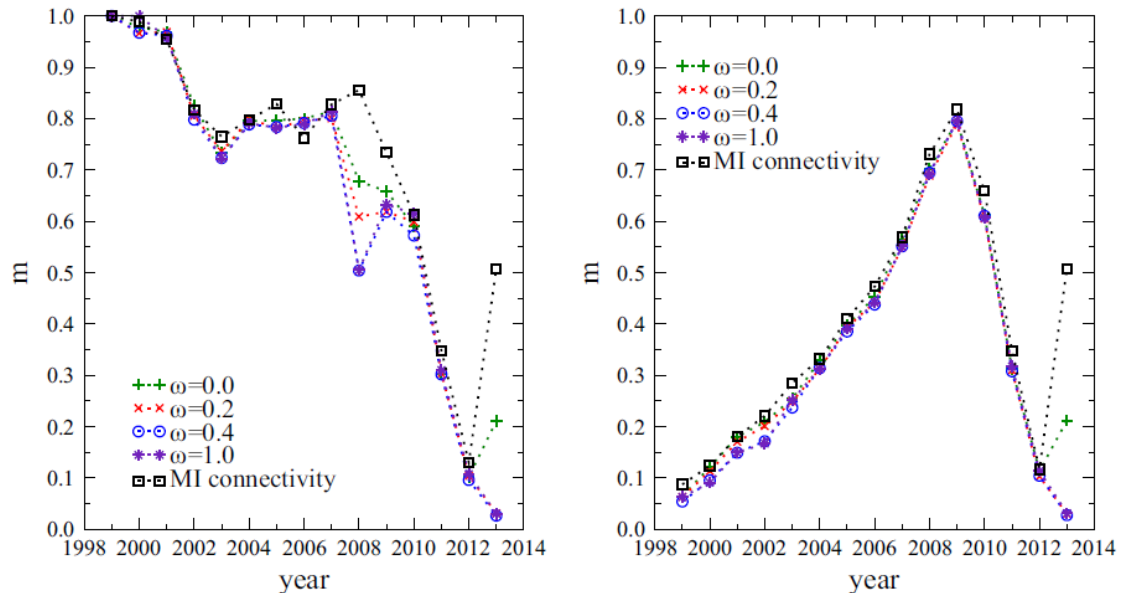


Figura 5-1: Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y constante de 0,5

Los valores más importantes son  $m$  y  $l$ .

En la gráfica, en la variable tenemos años por encima del valor 0,8 hasta el año 2007. Igualmente pasa lo mismo con  $l$ , sin embargo a diferencia de  $m$  en  $l$  el año 2006 está por debajo de 0,8. La duración del algoritmo es de aproximadamente 13 minutos. En Eclipse el programa ha llegado a un máximo de 20 megabytes durante su ejecución.



**Figura 5-2: Valores de  $m$  que demuestran los patrones recuperados con la red neuronal binaria. Usa los datos de  $N = 5897$ ,  $K = 200$  y  $t=100$ . A la izquierda está el resultado con el método online. A la derecha está el resultado con el método offline.**

Si comparamos las gráficas, en la primera gráfica podemos observar como los datos recuperados por encima de un umbral igual 0.8 llega hasta 2007, igual que en la segunda empieza a caer en 2007. La diferencia es que la primera solo cae cuando pasa de 2007, mientras que la segunda hay valores que bajan por debajo 0,8 durante los primeros años. Podemos concluir que la red neuronal ternaria tiene una mayor precisión que le permite reconocer los patrones, tanto la localización de los patrones reconocidos como que valores deben tener.

### 5.3 Mejoras

#### 5.3.1 Usa una constante C

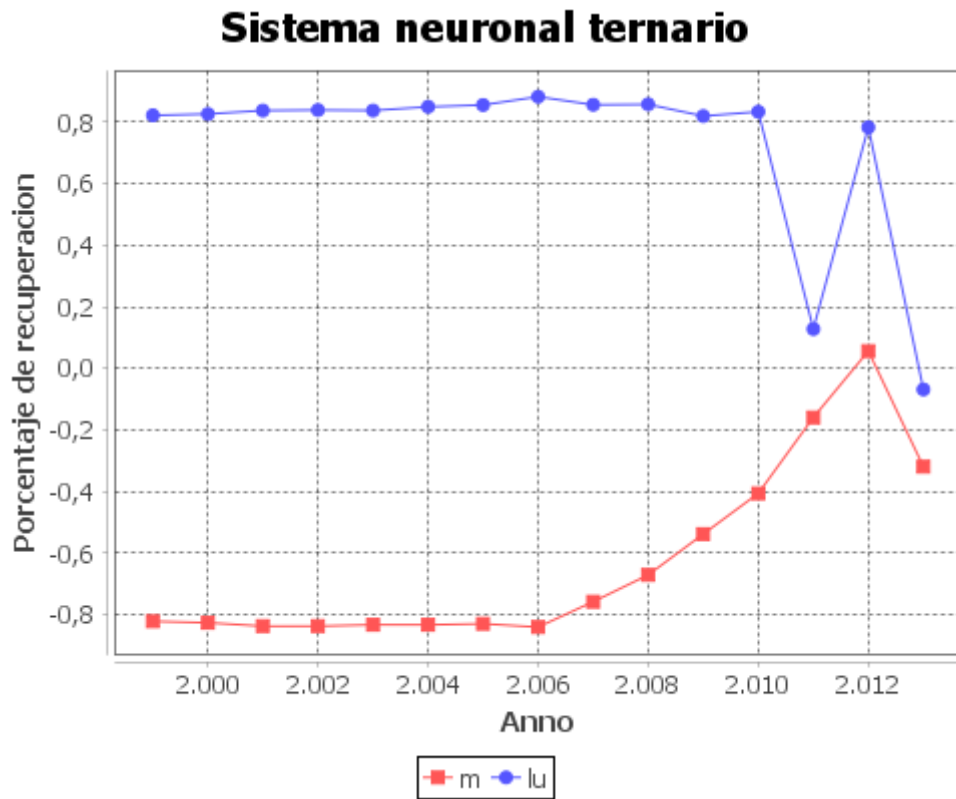
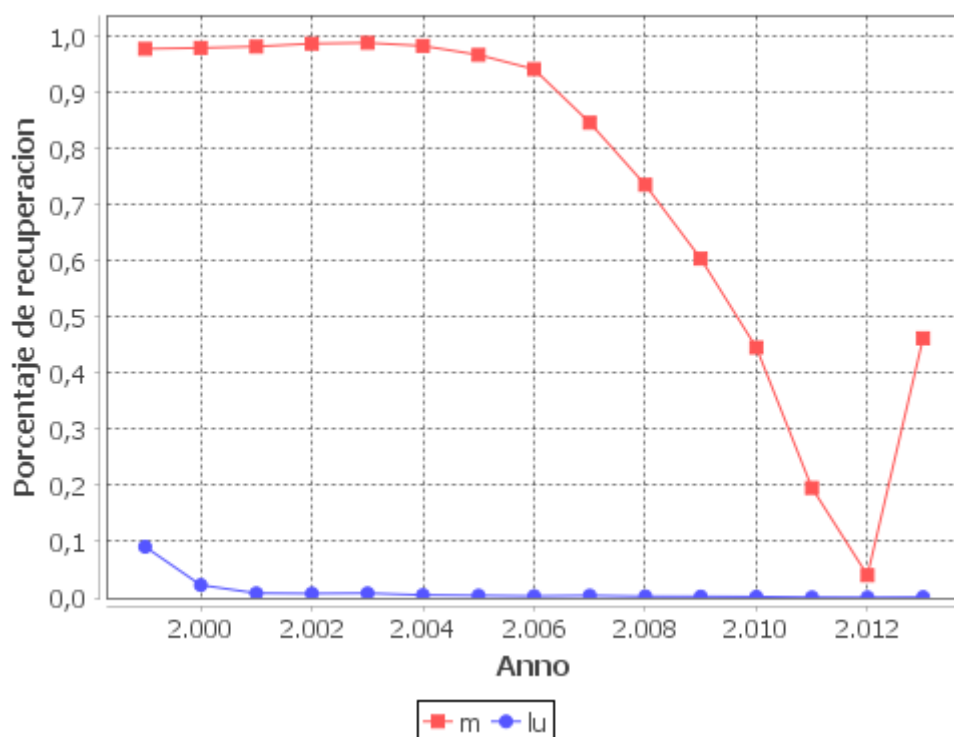


Figura 5-3: Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y con valor C de 0



## Sistema neuronal ternario

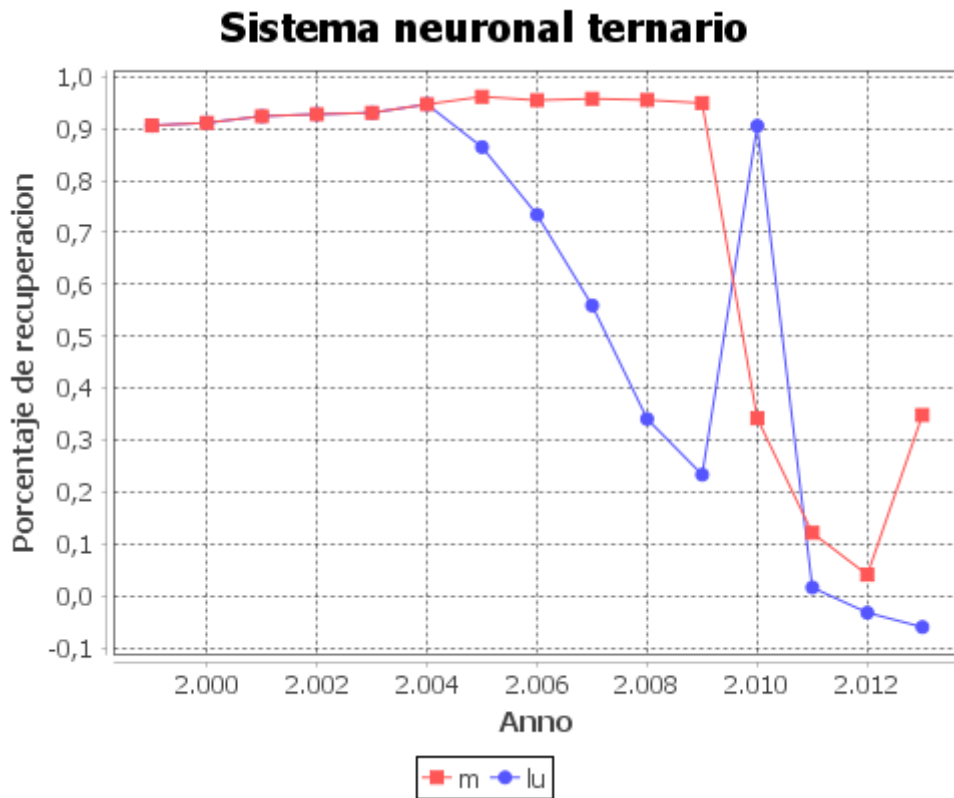


**Figura 5-4: Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y con un valor C de 1**

Como podemos ver el uso de una constante puede afectar al valor de las variables m y l. Por tanto podemos ajustar los valores de m y l para poder recuperar más años sin que ninguna de las 2 variables sufra demasiado ya que como vemos en los casos límites l puede acabar dando 0 y m dando valores negativos.

La constante como modifica el valor de J y K influye en la función de activación de la neurona donde si llega a uno de los extremos solo tiene K o J para hacer la función. Esto puede repercutir como que las neuronas nunca cambien a negativo por ejemplo.

### 5.3.2 Usar una matriz de patrones con pesos



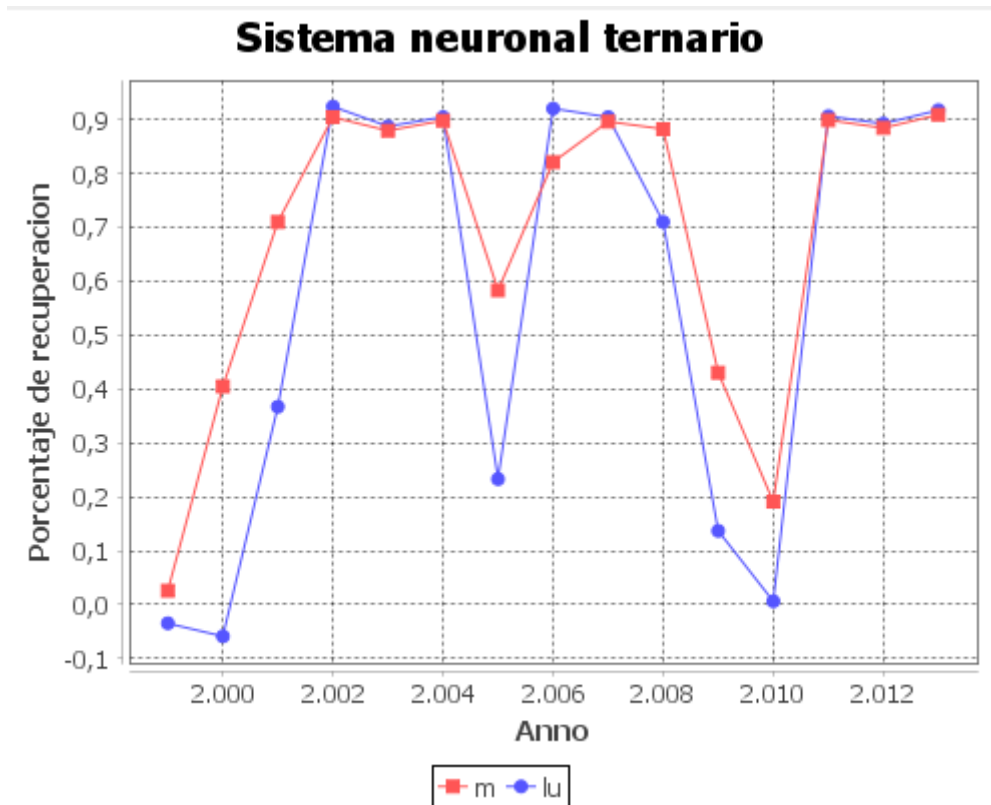
**Figura 5-5: Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y constante C de 0,5**

El uso de patrones con peso ha permitido mejorar los valores de m, pudiendo en este caso aumentar su eficacia hasta 2009 aprendiendo todos los años hasta allí para luego caer abruptamente hasta 0,35.

Esta mejora de resultados se debe a una mejor decisión del valor que tienen los patrones para cada año, puede hacer que los primeros años sean más relevantes al no tener que depender de un número de indicadores determinado para todos los años y por tanto tiene más posibilidades

## 5.4 Robustez

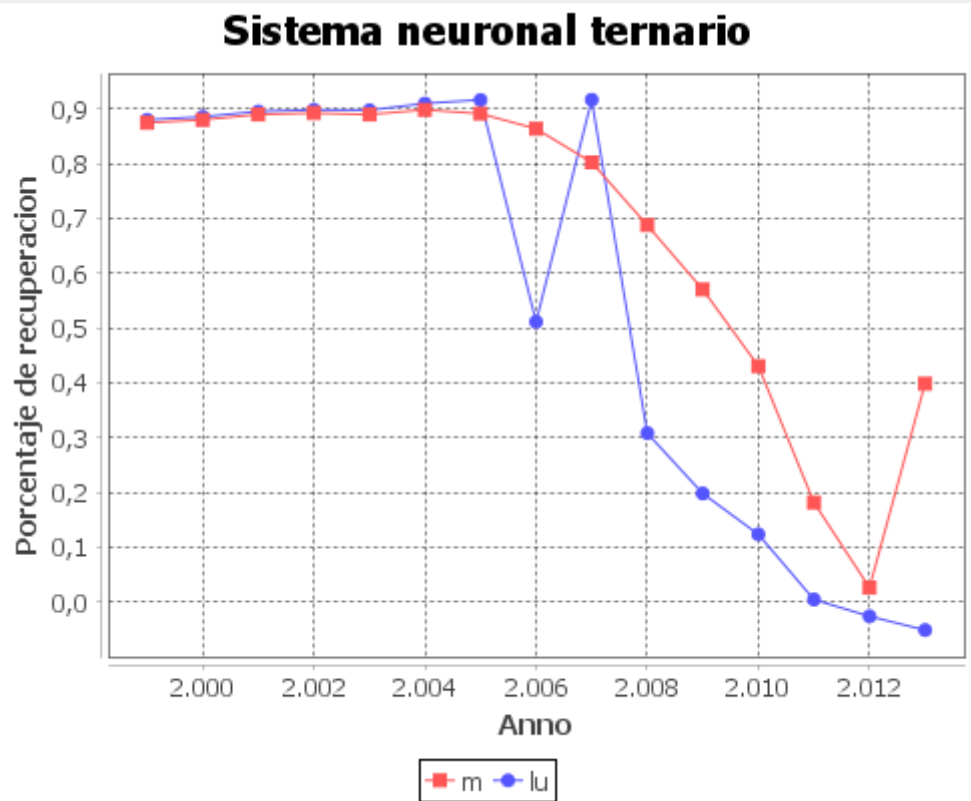
### 5.4.1 Capacidad de aprender de años desordenados



**Figura 5-6:** Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y constante C de 0,5 con los años desordenados

En la gráfica vemos que el desorden de los años hace que muestre una gráfica distinta, sin embargo podemos notar que los años desordenados se parecen los valores a la gráfica normal porque por ejemplo el número de años por encima del umbral 0,8 son el mismo que en la gráfica normal, y los resultados son muy parecidos por lo que se intuye que el entrenamiento de la gráfica puede aprender que años son los más adecuados para aprender del conjunto de datos.

### 5.4.2 Ruido capaz de soportar



**Figura 5-7:** Gráfica con los resultados del algoritmo con una muestra de 5897 empresas, en un lapso de 15 años y constante C de 0,5 con un ruido del 0,1

Aunque el ruido afecte a la red neuronal podemos ver que puede seguir aprendiendo años hasta 2007, usando de umbral 0.8, el valor que obtenemos aunque sin duda puede haberse afectado por el ruido, pero podemos ver que sigue siendo muy parecido a los valores de la gráfica normal, por lo que podemos concluir que a pesar de perder un poco en la recuperación de años aprendidos puede aguantar un porcentaje de ruido pequeños, si el ruido fuese mayor podríamos concluir que la recolección de datos ha sido errónea al incluir tantos errores.

Para demostrarlo hemos añadido un ruido del 10 %, un porcentaje que podríamos decir que es alto, el valor de los algoritmos de aprendizaje es que podamos usarlo con datos correctos y pueda reconocer patrones, si los patrones son erróneos podrían dar

# 6 Conclusiones y trabajo futuro

---

## 6.1 Conclusiones

Hemos podido probar que la nueva red tiene un mejor rendimiento para poder estudiar el informe de sostenibilidad GRI y por tanto ser de mayor utilidad en un futuro. Podremos reconocer patrones y poder establecer mejores relaciones según entre sectores y entre continentes.

Con la red ternaria hemos conseguido un sistema que evoluciona con un activador que se adapta. Para conseguir estos resultados hemos usado 3 parámetros, el solapamiento  $m$ , la actividad de patrones  $a$  y la parametrización  $l$  de la actividad neuronal  $q$  y la relación actividad-solapamiento  $n$  solapamiento  $m$ . Son muy positivos si los comparamos con la anterior ANN basada en una red neuronal binaria.

Podemos ver que cuando se van sumando cada vez más empresas que envían sus informes de sostenibilidad a GRI podemos llegar a un punto de saturación y necesitar por tanto otro periodo de aprendizaje hasta conseguir estabilidad para poder predecir con seguridad el comportamiento futuro de las empresas.

Las primeras empresas en enviar informes de sostenibilidad han demostrado lo positivo que es estar comprometido con comprometerse a mejorar su implicación tanto social como en el medio ambiente y presentar un informe serio sobre su situación económica para atraer a futuros inversores y obtener nuevos clientes.

Los intentos de mejorar el algoritmo con pequeñas modificaciones pueden resultar útiles. Añadir una constante para modificar los valores para el activador nos puede servir para mejorar la ganancia de información si encontramos la constante que mejor se comporta para la red, que pueda conseguir el máximo  $m$  sin perder demasiado de  $l$  por ejemplo. También hemos conseguido una mejoría en el aprendizaje de patrones si tomamos como referencia  $m$  que ha mejorado su rendimiento en 2 años respecto a la gráfica normal si usamos una matriz de patrones que hayamos obtenido al traducir la tabla de GRI usando distintos pesos a los años de la tabla según el número de indicadores que tiene.

Las redes neuronales por lo general son muy resistentes al ruido y las pruebas hechas con los años de los patrones desordenados han sido positivas por lo que el sistema es robusto y fiable para su utilización.

## 6.2 Trabajo futuro

Con esta red neuronal ternaria podremos estudiar mejor las empresas que realizan los informes GRI y su relación entre sectores y continentes. Esto podrá permitir estudiar si queremos en un futuro cómo evolucionan las empresas en los distintos apartados de GRI. En el caso de este TFG podremos estudiar mejor su evolución en la igualdad de género.

Conseguir un mayor rendimiento usando estas redes neuronales permitirá a las empresas que quieran invertir en un sector conocer mejor los países más sostenibles para poder invertir y que salga rentable.

Aun se puede mejorar la predicción de la sostenibilidad de las empresas, en cuyo caso será de gran ayuda para evitar una crisis como la actual o en caso de ocurrir reducir el periodo de recuperación.

Cada vez más empresas se encuentran concienciadas sobre demostrar con sus informes que quieren construir una nueva imagen basada en conductas positivas para la sociedad y el medio ambiente. Para poder adaptarse mejor a un mayor número de empresas y con cada vez más avances en el marco tanto económico como social y medio ambiente, GRI evoluciona sus guías para hacer informes de sostenibilidad para poder mantenerse como estándar de sostenibilidad. Con las nuevas guías será necesario adaptar la ANN para que aprenda mejor.



# Referencias

---

- [1] Three-State Neural Network: From mutual information to the Hamiltonian, David R. Carreta Dominguez
- [2] Modeling Sustainability report scoring sequences using an attractor network, David R. Carreta
- [3] Flow Diagrams of the Quadratic Neural Network, David R. Carreta Dominguez
- [4] The Antiquadrupolar Phase of the Biquadratic Neural Network, David R. Carreta Dominguez
- [5] Fundamentals of Neural Networks architecture, algorithms and applications, Laurence Fausett
- [6] G4 Sustainability Reporting Guidelines implementation manual <https://www.globalreporting.org/resourcelibrary/GRIG4-Part2-Implementation-Manual.pdf>, GRI
- [7] [https://es.wikipedia.org/wiki/Mapa\\_autoorganizado](https://es.wikipedia.org/wiki/Mapa_autoorganizado)
- [8] [https://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](https://es.wikipedia.org/wiki/Red_neuronal_artificial)
- [9] <https://es.wikipedia.org/wiki/Econom%C3%ADa>
- [10] <https://www.semanticscholar.org/paper/Image-Parsing-with-a-Three-State-Series-Neural-Seyedhosseini-Paiva/944ca9cfc53cfdcece2fa98e76d54a99452d7d65/pdf>
- [11] <https://chemazdamundi.wordpress.com/2010/08/22/falacias-matematicas-ii-la-importancia-de-las-matematicas-y-los-modelos-matematicos-en-la-economia-%C2%BFpor-que-utiliza-matematicas-la-economia-%C2%BFes-la-economia-una-ciencia-si-si-lo-es/>





## **Glosario**

---

GRI	Global Reporting Initiative
ANN	Attractor Neural Network

## **Anexos**

---

### ***A Manual de instalación***

Se recomienda usar Eclipse para usar el programa. Se importa la carpeta con el proyecto. Se tiene que re direccionar las referencias a las librerías externas, todas las necesarias están dentro del proyecto así que solo hace cambiar la dirección.

Para ejecutar la función main hace falta poner 4 argumentos. El orden es:

<Dirección del archivo> <número de empresas> <número de años> < Constante>



## B Manual del programador

### B.1 ExcelGRI.java

```
package tfg;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Iterator;

import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelGRI {

    public ExcelGRI(int filas, int columnas) {
        super();
        this.matriz = new double[filas][columnas];
    }

    private double[][] matriz;

    public double[][] readBinario(String direccion) throws IOException {
        String excelFilePath = direccion;
        //"griSEC13binary.xlsx"
        FileInputStream inputStream = new FileInputStream(new
File(excelFilePath));
        int inicio = 7;
        int current = 0;
        int fila = 0;

        Workbook workbook = new XSSFWorkbook(inputStream);
        Sheet firstSheet = workbook.getSheetAt(0);
        Iterator<Row> iterator = firstSheet.iterator();
        Row nextRow = iterator.next();

        while (iterator.hasNext()) {
            nextRow = iterator.next();
            Iterator<Cell> cellIterator = nextRow.cellIterator();
            current = 0;
            if (fila >= matriz.length)
                break;

            while (cellIterator.hasNext()) {
                if (current - inicio >= matriz[0].length)
                    break;

                Cell cell = cellIterator.next();
                String contenido = "";
                if (current >= inicio) {
                    if (cell == null) {
```

```

        contenido = "";
        matriz[fila][current - inicio] = 0;
    } else {

        switch (cell.getCellType()) {
        case Cell.CELL_TYPE_STRING:
            contenido =
cell.getStringCellValue();
            break;
        case Cell.CELL_TYPE_NUMERIC:
            contenido =
Double.toString(cell.getNumericCellValue());
            break;
        }
        if (contenido == "0")
            matriz[fila][current - inicio] =
0;
        else
            matriz[fila][current - inicio] =
1;
    }
}
current++;
}
fila++;
}

workbook.close();
inputStream.close();

return matriz;
}

public double[][] readTernario(String direccion) throws IOException {
    String excelFilePath = direccion;
    FileInputStream inputStream = new FileInputStream(new
File(excelFilePath));
    int inicio = 7;
    int current = 0;
    int fila = 0;

    Workbook workbook = new XSSFWorkbook(inputStream);
    Sheet firstSheet = workbook.getSheetAt(0);
    Iterator<Row> iterator = firstSheet.iterator();
    Row nextRow = iterator.next();

    while (iterator.hasNext()) {
        nextRow = iterator.next();
        Iterator<Cell> cellIterator = nextRow.cellIterator();
        current = 0;
        if (fila >= matriz.length)
            break;
        while (cellIterator.hasNext()) {

            if (current - inicio >= matriz[0].length)
                break;

            Cell cell = cellIterator.next();
            String contenido = "";

```

```

        if (current >= inicio) {
            if (cell == null) {
                contenido = "";
                matriz[fila][current - inicio] = -1;
            } else {

                switch (cell.getCellType()) {
                    case Cell.CELL_TYPE_STRING:
                        contenido =
cell.getStringCellValue();
                        contenido.split(",");

                        inicio] = 0;

                        inicio] = 1;

                        Double.toString(cell.getNumericCellValue());
                        0)
                        inicio] = -1;

                    case Cell.CELL_TYPE_NUMERIC:
                        contenido =
                        if (cell.getNumericCellValue() ==
                                matriz[fila][current -
                                        break;
                }
            }
            current++;
        }
        fila++;
    }

    workbook.close();
    inputStream.close();

    return matriz;
}

public double[][] readTernarioPeso(String direccion) throws IOException {
    String excelFilePath = direccion;
    FileInputStream inputStream = new FileInputStream(new
File(excelFilePath));
    int inicio = 7;
    int current = 0;
    int fila = 0;
    double[][] matrizSecundaria = new
double[matriz.length][matriz[0].length];

    Workbook workbook = new XSSFWorkbook(inputStream);
    Sheet firstSheet = workbook.getSheetAt(0);
    Iterator<Row> iterator = firstSheet.iterator();
    Row nextRow = iterator.next();

    while (iterator.hasNext()) {
        if (fila >= matrizSecundaria.length)
            break;

```

```

nextRow = iterator.next();
Iterator<Cell> cellIterator = nextRow.cellIterator();
current = 0;

while (cellIterator.hasNext()) {

    if (current - inicio >= matrizSecundaria[0].length)
        break;
    Cell cell = cellIterator.next();
    String contenido = "";
    if (current >= inicio) {
        if (cell == null) {
            contenido = "";
            matrizSecundaria[fila][current - inicio]
= 0;
        } else {

            switch (cell.getCellType()) {
                case Cell.CELL_TYPE_STRING:
                    contenido =
cell.getStringCellValue();
                    String palabras[] =
contenido.split(",");
                    matrizSecundaria[fila][current -
inicio] = palabras.length;
                    break;
                case Cell.CELL_TYPE_NUMERIC:
                    contenido =
Double.toString(cell.getNumericCellValue());
                    if (cell.getNumericCellValue() ==
0)
matrizSecundaria[fila][current - inicio] = 0;
                    break;
            }
        }
        current++;
    }
    fila++;
}

workbook.close();
inputStream.close();
double media = 0.0;
double sesgo = 0.0;
for (int j = 0; j < matriz[0].length; j++) {
    for (int i = 0; i < matriz.length; i++) {
        media += matrizSecundaria[i][j];
    }
    media = media / matriz.length;
    sesgo = 2 * (media / 3);
    for (int i = 0; i < matriz.length; i++) {
        if (sesgo == 0)
            matriz[i][j] = -1;
        else if (sesgo > 0 && sesgo <= 0.5) {
            if (matrizSecundaria[i][j] < sesgo)
                matriz[i][j] = -1;
            else

```



```

        matriz[i][j] = 0;
    } else {
        if (matrizSecundaria[i][j] < sesgo)
            matriz[i][j] = -1;
        else if (matrizSecundaria[i][j] >= sesgo &&
matrizSecundaria[i][j] < (sesgo * 2))
            matriz[i][j] = 0;
        else
            matriz[i][j] = 1;
    }
    }
}
return matriz;
}
}

public double[][] getMatriz() {
    return matriz;
}
}
}

```

## B.2 MathsTernario.java

```

package tfg;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;

public class MathsTernario {

    private int[] Neuronas;
    private double[][] Patrones;
    private double[][] nu;
    private double ActividadMedia;
    private int K;
    private int P;
    private double C;
    private double[][] KK;
    private double[][] JJ;
    private double[] Actividades;

    public MathsTernario(int filas, int columnas, double constante) {

        Patrones = new double[filas][columnas];
        nu = new double[filas][columnas];
        KK = new double[filas][filas];
        JJ = new double[filas][filas];
        Actividades = new double[columnas];
        ActividadMedia = 0.0;
        // TODO Auto-generated constructor stub
        P = columnas;
        K = filas;
        C = constante;
    }
}

```

```

        Neuronas = new int[K];
    }

    public void iniciliarizarNeuronas(int anno) {
        for (int i = 0; i < Neuronas.length; i++)
            Neuronas[i] = (int) Patrones[i][anno];
    }

    public double m(int anno) {
        double res = 0.0;

        for (int i = 0; i < Neuronas.length; i++)
            res += Neuronas[i] * Patrones[i][anno];
        // return res/(neuronas.length);
        return res / (Actividades[anno] * Neuronas.length);
    }

    public double q(int anno) {
        double res = 0.0;

        for (int i = 0; i < Neuronas.length; i++)
            res += Math.pow(Neuronas[i], 2);
        res = res / Neuronas.length;
        return res;
    }

    public double n(int anno) {
        double res = 0.0;

        for (int i = 0; i < Neuronas.length; i++)
            res += Math.pow(Neuronas[i], 2) *
Math.pow(Patrones[i][anno], 2);
        // res = res/(neuronas.length);
        res = res / (Actividades[anno] * Neuronas.length);
        return res;
    }

    public double I(double m, double lu, double q, int anno) {
        double res = (1 / 2.0) * (Actividades[anno] / q) * Math.pow(m, 2);
        res += (1 / 2.0) * ((Actividades[anno] * (1 - Actividades[anno])) /
(q * (1 - q))) * Math.pow(lu, 2);

        return res;
    }

    public double lu(int anno) {
        return (n(anno) - q(anno)) / (1 - Actividades[anno]);
    }

    public double H() {
        return H1() + H2();
    }

    public double H1() {
        double res = 0.0;
        for (int i = 0; i < Neuronas.length; i++) {
            for (int j = 0; j < Neuronas.length; j++) {

```

```

        res += JJ[i][j] * Neuronas[i] * Neuronas[j];
    }
}
res = res / 2;
return -res;
}

public double H2() {
    double res = 0.0;
    for (int i = 0; i < Neuronas.length; i++) {
        for (int j = 0; j < Neuronas.length; j++) {
            res += KK[i][j] * Math.pow(Neuronas[i], 2) *
Math.sqrt(Neuronas[j]);
        }
    }
    res = res / 2;
    return -res;
}

public void J(int i, int j) {
    double res = 0.0;
    for (int anno = 0; anno < P; anno++)
        res += Patrones[i][anno] * Patrones[j][anno];
    res = res * (C);
    JJ[i][j] = res / (Math.pow(ActividadMedia, 2) * K);
}

public void K(int i, int j) {
    double res = 0.0;
    for (int anno = 0; anno < P; anno++)
        res += nuu(i, anno) * nuu(j, anno);
    res = res * (1 - C);
    KK[i][j] = res / K;
}

public void J() {
    for (int i = 0; i < K; i++)
        for (int j = 0; j < K; j++)
            J(i, j);
}

public void K() {
    for (int i = 0; i < K; i++)
        for (int j = 0; j < K; j++)
            K(i, j);
}

public double nuu(int i, int anno) {
    return (Math.pow(Patrones[i][anno], 2) - Actividades[anno]) /
(Actividades[anno] * (1 - Actividades[anno]));
}

public void funcionActivacion(int indice) {
    double oit = activadorit(indice);
    double hit = potencialNeuronait(indice);
}

```

```

        Neuronas[indice] = signo(hit) * G(Math.sqrt(Math.pow(hit, 2)) +
oit);
    }

    public int signo(double h) {
        if (h < 0)
            return -1;
        else
            return 1;
    }

    public int G(double h) {
        if (h > 0)
            return 1;
        else
            return 0;
    }

    // hit
    public double potencialNeuronait(int i) {
        double res = 0.0;
        for (int j = 0; j < K; j++) {
            res += JJ[i][j] * Neuronas[j];
        }
        return res;
    }

    // oit
    public double activadorit(int i) {
        double res = 0.0;
        for (int j = 0; j < K; j++) {
            res += KK[i][j] * Math.pow(Neuronas[j], 2);
        }
        return res;
    }

    public void actividades0(){
        double activos=0.0;
        for(int i = 0;i<P;i++){
            for(int j = 0;j<K;j++)
            {
                activos += Math.pow(Patrones[j][i],2);
            }
            Actividades[i] = activos/K;
            activos=0.0;
        }
        double res = 0.0;
        for(int i = 0;i<P;i++)
            res+=Actividades[i];
        ActividadMedia = res/P;
    }

    public void actividadesConstante(){
        for(int i = 0;i<P;i++){

            Actividades[i] = 0.6;
        }
    }

```

```

        ActividadMedia = 0.6;
    }

    public static void main(String[] args) {

        if(args.length!= 4){
            System.out.print("Numero de argumentos incorrectos");
            return;
        }
        String direccion = args[0];
        int empresas = Integer.parseInt(args[1]);
        int annos = Integer.parseInt(args[2]);
        double constante = Double.parseDouble(args[3]);
        ExcelGRI excel = new ExcelGRI(empresas, annos);
        MathsTernario ternario = new MathsTernario(empresas, annos,
constante);

        ArrayList<ArrayList<Double>> listas = new
ArrayList<ArrayList<Double>>();
        ArrayList<Double> m = new ArrayList<Double>(), n = new
ArrayList<Double>(), q = new ArrayList<Double>(),
            H = new ArrayList<Double>();
        ArrayList<Double> lu = new ArrayList<Double>();
        ArrayList<Double> I = new ArrayList<Double>();
        try {
            ternario.setPatrones(excel.readTernario(direccion));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Date fin;
        ternario.actividades0();
        ternario.K();
        ternario.J();
        for (int i = 0; i < ternario.getP(); i++) {
            ternario.inicializarNeuronas(i);
            for (int b = 0; b < 10; b++)
                for (int j = 0; j < ternario.getK(); j++) {
                    ternario.funcionActivacion(j);
                }
            m.add(ternario.m(i));
            //n.add(ternario.n(i));
            //q.add(ternario.q(i));
            // H.add(ternario.H());
            lu.add(ternario.lu(i));
            //I.add(ternario.I(m.get(i), lu.get(i), q.get(i),i));
        }
        fin = new java.util.Date();
        System.out.println("finaliza en" + fin);
        listas.add(m);
        //listas.add(n);
        //listas.add(q);
        listas.add(lu);
        //listas.add(I);
        Diagrama diagrama = new Diagrama(listas);
        Interfaz frame = new Interfaz(diagrama);
        frame.setVisible(true);
    }
}

```

```

}

public int[] getNeuronas() {
    return Neuronas;
}

public void setNeuronas(int[] neuronas) {
    this.Neuronas = neuronas;
}

public double[][] getPatrones() {
    return Patrones;
}

public void setPatrones(double[][] patrones) {
    this.Patrones = patrones;
}

public void setPatronesDesordenados(double[][] patrones) {
    ArrayList<Integer> indices = new ArrayList<Integer>();
    for (int i = 0; i < patrones[0].length; i++) {
        indices.add(i);
    }

    Collections.shuffle(indices);
    for (int i = 0; i < patrones.length; i++)
        for (int j = 0; j < patrones[0].length; j++) {
            this.Patrones[i][j] = patrones[i][indices.get(j)];
        }
}

public void setPatronesRuido(double[][] patrones) {
    double ini = 0;
    double med = 0;
    double fin = 0;
    double random = 0;
    for (int j = 0; j < patrones[0].length; j++) {
        for (int i = 0; i < patrones.length; i++) {
            if (patrones[i][j] == -1)
                ini++;
            else if (patrones[i][j] == 0)
                med++;
            else
                fin++;
        }
        for (int i = 0; i < patrones.length; i++) {
            random = Math.random();
            if (random <= 0.1 * ini / patrones.length) {
                this.Patrones[i][j] = -1;
            } else if (random > 0.1 * ini / patrones.length
+ ini / patrones.length)
                this.Patrones[i][j] = 0;
            else if (random > 0.1 * (med / patrones.length + ini /
patrones.length) && random <= 0.1)
                this.Patrones[i][j] = 1;
            else
                this.Patrones[i][j] = patrones[i][j];
        }
}

```

```

        ini = 0;
        med = 0;
        fin = 0;
    }
}

public double[][] getNu() {
    return nu;
}

public void setNu(double[][] nu) {
    this.nu = nu;
}

public double getActividad() {
    return ActividadMedia;
}

public double getP() {
    return P;
}

public double getK() {
    return K;
}

public void setActividad(double actividad) {
    this.ActividadMedia = actividad;
}
}

```

### B.3 Diagrama.java

```
package tfg;
```

```
import java.awt.Color;
import java.awt.image.BufferedImage;
import java.util.ArrayList;
```

```
import javax.swing.ImageIcon;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.labels.StandardXYItemLabelGenerator;
import org.jfree.chart.labels.XYItemLabelGenerator;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYSeries;
```

```

import org.jfree.data.xy.XYSeriesCollection;

public class Diagrama extends ImageIcon{

    public Diagrama(ArrayList<ArrayList<Double>> listas){

        //se declara el grafico XY Lineal
        XYDataset xydataset = xyDataset(listas);
        JFreeChart jfreechart = ChartFactory.createXYLineChart(
            "Sistema neuronal ternario", "Anno", "Porcentaje de recuperacion",
            xydataset, PlotOrientation.VERTICAL, true, false, false);

        //personalización del grafico
        XYPlot xyplot = (XYPlot) jfreechart.getPlot();
        xyplot.setBackgroundPaint( Color.white );
        xyplot.setDomainGridlinePaint( Color.BLACK );
        xyplot.setRangeGridlinePaint( Color.BLACK );
        // -> Pinta Shapes en los puntos dados por el XYDataset
        XYLineAndShapeRenderer xylineandshaperenderer = (XYLineAndShapeRenderer)
xyplot.getRenderer();
        xylineandshaperenderer.setBaseShapesVisible(true);
        //--> muestra los valores de cada punto XY
        XYItemLabelGenerator xy = new StandardXYItemLabelGenerator();
        xylineandshaperenderer.setBaseItemLabelGenerator( xy );
        xylineandshaperenderer.setBaseLinesVisible(true);
        //fin de personalización

        //se crea la imagen y se asigna a la clase ImageIcon
        BufferedImage bufferedImage = jfreechart.createBufferedImage( 500, 400);
        this.setImage(bufferedImage);
    }

    private XYDataset xyDataset(ArrayList<ArrayList<Double>> listas)
    {
        //se declaran las series y se llenan los datos
        XYSeries serie1 = new XYSeries("m");
        //serie #1
        for(int i = 0; i< listas.get(0).size();i++)
            serie1.add( i + 1999, listas.get(0).get(i));
        // XYSeries serie2 = new XYSeries("n");
        //serie #2
        //for(int i = 0; i< listas.get(1).size();i++)
        // serie2.add( i + 1999, listas.get(1).get(i));
        // XYSeries serie3 = new XYSeries("q");
        //serie #3
        //for(int i = 0; i< listas.get(2).size();i++)
        // serie3.add( i + 1999, listas.get(2).get(i));
        XYSeries serie4 = new XYSeries("lu");
        //serie #4
    }
}

```



```

for(int i = 0; i< listas.get(1).size();i++)
    serie4.add( i + 1999, listas.get(1).get(i));
//XYSeries serie5 = new XYSeries("I");
//serie #5
// for(int i = 0; i< listas.get(3).size();i++)
//  serie5.add( i + 1999, listas.get(4).get(i));

XYSeriesCollection xyseriescollection = new XYSeriesCollection();
xyseriescollection.addSeries( serie1 );
// xyseriescollection.addSeries( serie2 );
//xyseriescollection.addSeries( serie3 );
xyseriescollection.addSeries( serie4 );
// xyseriescollection.addSeries( serie5 );

return xyseriescollection;
}
}

```

#### B.4 Interfaz.java

```

package tfg;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Frame;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

public class Interfaz extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Interfaz frame = new Interfaz(null);

```

```

        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    });
}

/**
 * Create the frame.
 */
public Interfaz(Diagrama diag) {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 600,600);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);

    JPanel panel = new JPanel();
    contentPane.add(panel, BorderLayout.CENTER);
    panel.setLayout(new BorderLayout(0, 0));

    panel.setPreferredSize(new Dimension(500, 400));

    JLabel jLabel1 = new JLabel("label1");
    panel.add(jLabel1, BorderLayout.CENTER);

    if(diag == null)
        jLabel1.setIcon( new Diagrama(null));
    else
        jLabel1.setIcon(diag);
    jLabel1.setText("");
}
}
}

```

**C Anexo ...**