



## Eloise, Red Neuronal

---

# ELOISE

## Índice

Introducción..... 2.

### **Sección 1**

Preliminares ..... 4.

Redes Neuronales Artificiales ..... 4.

Proben1 ..... 16.

### **Sección 2**

*Algoritmos* ..... 23.

### **Sección 3**

Experimentación .....26.

*Conclusión* ..... 35.

*Bibliografía* ..... 35.



# Eloise, Red Neuronal

---

## Eloise

Alfonso Gastalver Llamas

Alumno de Inteligencia Artificial 2, Ingeniería Informática  
Universidad de Sevilla  
alfonsogastalverllamas@gmail.com

**Abstract.** El objetivo principal de este proyecto es el de realizar pruebas mediante una red neuronal sobre los ejemplos otorgados por PROBEN1[1], un conjunto de entrenamiento con ejemplos sobre un determinado tipo para ser clasificados. Las pruebas incluirán la clasificación correcta y un conjunto de resultados que deberán ser mostrados al usuario.

## 1 Introducción

Eloise es un proyecto implementado en java que crea una red neuronal y trabaja a raíz de un conjunto de ejemplos dados para clasificarlos correctamente.

Empezaremos describiendo principalmente ¿Qué es una red neuronal? ¿Para qué se utilizan? Seguiremos con PROBEN1 y finalizaremos con el objetivo principal.

¿Qué es una red neuronal?

Para entender este concepto es preciso hablar primero de la inspiración biológica. El campo de las redes neuronales surgió a raíz del aprendizaje biológico de la mente humana, y en concreto de las neuronas. Por tanto este va a ser el contexto en el que nos movamos.

“Las neuronas son un tipo de células del sistema nervioso cuya principal función es la excitabilidad eléctrica de su membrana plasmática; están especializadas en la recepción de estímulos y conducción del impulso nervioso (en forma de potencial de acción) entre ellas o con otros tipos celulares, como por ejemplo las fibras musculares de la placa motora.

Altamente diferenciadas, la mayoría de las neuronas no se dividen una vez alcanzada su madurez; no obstante, una minoría sí lo hace. Las neuronas presentan unas características morfológicas típicas que sustentan sus funciones: un cuerpo celular llamado soma o «pericarion», central; una o varias prolongaciones cortas que generalmente transmiten impulsos hacia el soma celular, denominadas dendritas; y una prolongación larga, denominada axón o «cilindroeje», que conduce los impulsos desde el soma hacia otra neurona u órgano diana.



## Eloise, Red Neuronal

---

Las neuronas transmiten ondas de naturaleza eléctrica originadas como consecuencia de un cambio transitorio de la permeabilidad en la membrana plasmática. Su propagación se debe a la existencia de una diferencia de potencial o potencial de membrana (que surge gracias a las concentraciones distintas de iones a ambos lados de la membrana, según describe el potencial de Nernst) entre la parte interna y externa de la célula (por lo general de  $-70$  mV). La carga de una célula inactiva se mantiene en valores negativos (el interior respecto al exterior) y varía dentro de unos estrechos márgenes. Cuando el potencial de membrana de una célula excitable se despolariza más allá de un cierto umbral (de  $65$  mV a  $55$  mV) la célula genera (o dispara) un potencial de acción. Un potencial de acción es un cambio muy rápido en la polaridad de la membrana de negativo a positivo y vuelta a negativo, en un ciclo que dura unos milisegundos.” [4]

Una red neuronal por tanto es un conjunto de neuronas que transmiten impulsos nerviosos entre ellas para realizar tareas. La red neuronal humana es la base de nuestro sistema nervioso. La rapidez se basa en el paralelismo masivo.

Basadas en estos aspectos biológicos, surgieron las redes neuronales artificiales, que suponen una extrapolación de dicho mecanismo biológico hacia un marco computacional.

¿Para qué sirven las redes neuronales artificiales?

Las redes neuronales artificiales nos permiten tomar decisiones. Y su uso está más extendido a problemas de clasificación, pues funcionan increíblemente bien con éstos. Y es el uso que nosotros le daremos en este proyecto

### PROBEN1

Proben1 es un proyecto informático realizado por Lutz Prechelt de la facultad de informática de Karlsruhe, Alemania. Se basa en una colección de problemas de aprendizaje mediante redes neuronales para clasificar sus respectivos conjuntos de ejemplos correctamente, con el objetivo de que los desarrolladores de redes neuronales puedan testearlas fácilmente.

El objetivo principal de este proyecto es por tanto el de clasificar correctamente y mostrar los resultados, varios de los conjuntos de entrenamiento de PROBEN1.

Para finalizar esta introducción, expondremos la estructura de este documento. En la sección 2: Preliminares, explicaremos el funcionamiento de las redes neuronales, y también profundizaremos sobre en Proben1. Después describiremos la implementación realizada del algoritmo. En otro apartado mostraremos los resultados obtenidos de los ejemplos evaluados usando Eloise, con diferentes redes neuronales



## Eloise, Red Neuronal

---

cada uno. El documento acaba con una conclusión sobre los datos obtenidos y la relevancia que éstos puedan darnos en un futuro.

### Preliminares

#### 1.1 Redes Neuronales Artificiales

Para ambientarnos comenzaremos adentrándonos en el marco de la computación neuronal con un poco de historia, y a partir de ahí explicaremos qué son, para qué se usan y que características y ventajas poseen las redes neuronales artificiales.

##### “Introducción a la Computación Neuronal

El hombre se ha caracterizado siempre por su búsqueda constante de nuevas vías para mejorar sus condiciones de vida. Estos esfuerzos le han servido para reducir el trabajo en aquellas operaciones en las que la fuerza juega un papel primordial. Los progresos obtenidos han permitido dirigir estos esfuerzos a otros campos, como por ejemplo, a la construcción de máquinas calculadoras que ayuden a resolver de forma automática y rápida determinadas operaciones que resultan tediosas cuando se realizan a mano.

Charles Babbage trató de construir una máquina capaz de resolver problemas matemáticos. Posteriormente otros tantos intentaron construir máquinas similares, pero no fue hasta la Segunda Guerra Mundial, cuando ya se disponía de instrumentos electrónicos, que se empezaron a recoger los primeros frutos. En 1946 se construyó la primera computadora electrónica, ENIAC. Desde entonces los desarrollos en este campo han tenido un auge espectacular.

Estas máquinas permiten implementar fácilmente algoritmos para resolver multitud de problemas que antes resultaban engorrosos de resolver. Sin embargo, se observa una limitación importante: ¿qué ocurre cuando el problema que se quiere resolver no admite un tratamiento algorítmico, como es el caso, por ejemplo, de la clasificación de objetos por rasgos comunes? Este ejemplo demuestra que la construcción de nuevas máquinas más versátiles requiere un enfoque del problema desde otro punto de vista. Los desarrollos actuales de los científicos se dirigen al estudio de las capacidades humanas como una fuente de nuevas ideas para el diseño de las nuevas máquinas. Así, la inteligencia artificial es un intento por descubrir y describir aspectos de la inteligencia humana que pueden ser simulados mediante máquinas. Esta disciplina se ha desarrollado fuertemente en los últimos años teniendo aplicación en algunos campos como visión artificial, demostración de teoremas, procesamiento de información expresada mediante lenguajes humanos... etc.



## Eloise, Red Neuronal

---

Las redes neuronales son más que otra forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Si se examinan con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se observará que todos ellos tienen una característica en común: la experiencia. El hombre es capaz de resolver estas situaciones acudiendo a la experiencia acumulada. Así, parece claro que una forma de aproximarse al problema consista en la construcción de sistemas que sean capaces de reproducir esta característica humana.

En definitiva, las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto del que disponemos para un sistema que es capaz de adquirir conocimiento a través de la experiencia. Una red neuronal es "un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: *la neurona*".

Todos los procesos del cuerpo humano se relacionan en alguna u otra forma con la (in)actividad de estas neuronas. Las mismas son un componente relativamente simple del ser humano, pero cuando millares de ellas se conectan en forma conjunta se hacen muy poderosas.

También, es bien conocido que los humanos son capaces de aprender. Aprendizaje significa que aquellos problemas que inicialmente no pueden resolverse, pueden ser resueltos después de obtener más información acerca del problema.

Por lo tanto, las Redes Neuronales:

- Consisten de unidades de procesamiento que intercambian datos o información.
- Se utilizan para reconocer patrones, incluyendo imágenes, manuscritos y secuencias de tiempo, tendencias financieras.
- Tienen capacidad de aprender y mejorar su funcionamiento.

Una primera clasificación de los modelos de redes neuronales podría ser, atendiendo a su similitud con la realidad biológica:

1) El modelo de tipo biológico. Este comprende las redes que tratan de simular los sistemas neuronales biológicos, así como las funciones auditivas o algunas funciones básicas de la visión.

2) El modelo dirigido a aplicación. Este modelo no tiene por qué guardar similitud con los sistemas biológicos. Su arquitectura está fuertemente ligada a las necesidades de las aplicaciones para la que es diseñada.



## Eloise, Red Neuronal

---

### Historia de las Redes Neuronales

Conseguir diseñar y construir máquinas capaces de realizar procesos con cierta inteligencia ha sido uno de los principales objetivos de los científicos a lo largo de la historia. De los intentos realizados en este sentido se han llegado a definir las líneas fundamentales para la obtención de máquinas inteligentes: En un principio los esfuerzos estuvieron dirigidos a la obtención de autómatas, en el sentido de máquinas que realizaran, con más o menos éxito, alguna función típica de los seres humanos. Hoy en día se continúa estudiando en ésta misma línea, con resultados sorprendentes, existen maneras de realizar procesos similares a los inteligentes y que podemos encuadrar dentro de la llamada Inteligencia Artificial (IA).

A pesar de disponer de herramientas y lenguajes de programación diseñados expresamente para el desarrollo de máquinas inteligentes, existe un enorme problema que limita los resultados que se pueden obtener: estas máquinas se implementan sobre computadoras basadas en la filosofía de Von Neumann, y que se apoyan en una descripción secuencial del proceso de tratamiento de la información. Si bien el desarrollo de estas computadoras es espectacular, no deja de seguir la línea antes expuesta: una máquina que es capaz de realizar tareas mecánicas de forma increíblemente rápida, como por ejemplo cálculo, ordenación o control, pero incapaz de obtener resultados aceptables cuando se trata de tareas como reconocimiento de formas, voz, etc.

La otra línea de la investigación ha tratado de aplicar principios físicos que rigen en la naturaleza para obtener máquinas que realicen trabajos pesados en nuestro lugar. De igual manera se puede pensar respecto a la forma y capacidad de razonamiento humano; se puede intentar obtener máquinas con esta capacidad basadas en el mismo principio de funcionamiento.

No se trata de construir máquinas que compitan con los seres humanos, sino que realicen ciertas tareas de rango intelectual con que ayudarle, principio básico de la Inteligencia Artificial.

Las primeras explicaciones teóricas sobre el cerebro y el pensamiento ya fueron dadas ya por Platón (427-347 a.C.) y Aristóteles (348-422 a.C.). Las mismas ideas también las mantuvo Descartes (1569-1650) y los filósofos empiristas del siglo XVIII.

La clase de las llamadas *máquinas cibernéticas*, a la cual la computación neuronal pertenece, tiene más historia de la que se cree: Herón (100 a.C) construyó un autómata hidráulico.

1936 - Alan Turing. Fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los



## Eloise, Red Neuronal

---

fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas (Un Cálculo Lógico de la Inminente Idea de la Actividad Nerviosa - Boletín de Matemática Biofísica 5: 115-133). Ellos modelaron una red neuronal simple mediante circuitos eléctricos.

1949 - Donald Hebb. Escribió un importante libro: *La organización del comportamiento*, en el que se establece una conexión entre psicología y fisiología. Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aun hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados. También intentó encontrar semejanzas entre el aprendizaje y la actividad nerviosa. Los trabajos de Hebb formaron las bases de la Teoría de las Redes Neuronales.

1950 - Karl Lashley. En sus series de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro sino que era distribuida encima de él.

1956 - Congreso de Dartmouth. Este Congreso frecuentemente se menciona para indicar el nacimiento de la inteligencia artificial.

1957 - Frank Rosenblatt. Comenzó el desarrollo del Perceptrón. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como reconocedor de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado anteriormente. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente. En 1959, escribió el libro Principios de Neurodinámica, en el que confirmó que, bajo ciertas condiciones, el aprendizaje del Perceptrón convergía hacia un estado finito (Teorema de Convergencia del Perceptrón).

1960 - Bernard Widrow/Marcial Hoff. Desarrollaron el modelo Adaline (ADAPtative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.

1961 - Karl Steinbeck: Die Lernmatrix. Red neuronal para simples realizaciones técnicas (memoria asociativa).



## Eloise, Red Neuronal

---

1967 - Stephen Grossberg. A partir de sus conocimientos fisiológicos, ha escrito numerosos libros y desarrollado modelo de redes neuronales. Realizó una red: Avalancha, que consistía en elementos discretos con actividad que varía en el tiempo que satisface ecuaciones diferenciales continuas, para resolver actividades como reconocimiento continuo de habla y aprendizaje de los brazos de un robot.

1969 - Marvin Minsky/Seymour Papert. En este año surgieron críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. Minsky y Papert, del Instituto Tecnológico de Massachusetts (MIT), publicaron un libro *Perceptrons*. Probaron (matemáticamente) que el Perceptrón no era capaz de resolver problemas relativamente fáciles, tales como el aprendizaje de una función no-lineal. Esto demostró que el Perceptrón era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real. A pesar del libro, algunos investigadores continuaron su trabajo. Tal fue el caso de James Anderson, que desarrolló un modelo lineal, llamado Asociador Lineal, que consistía en unos elementos integradores lineales (neuronas) que sumaban sus entradas. Este modelo se basa en el principio de que las conexiones entre neuronas son reforzadas cada vez que son activadas. Anderson diseñó una potente extensión del Asociador Lineal, llamada Brain State in a Box (BSB).

1974 - Paul Werbos. Desarrolló la idea básica del algoritmo de aprendizaje de *propagación hacia atrás* (backpropagation); cuyo significado quedó definitivamente aclarado en 1985.

1977 - Stephen Grossberg. Teoría de Resonancia Adaptada (TRA). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo.

1977 - Teuvo Kohonen. Ingeniero electrónico de la Universidad de Helsinki, desarrolló un modelo similar al de Anderson, pero independientemente.

1980 - Kunihiko Fukushima. Desarrolló un modelo neuronal para el reconocimiento de patrones visuales..

1985 - John Hopfield. Provocó el renacimiento de las redes neuronales con su libro: "Computación neuronal de decisiones en problemas de optimización."

1986 - David Rumelhart/G. Hinton. Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation). A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales. En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las





## Eloise, Red Neuronal

---

aplicaciones nuevas que surgen (sobre todo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación).

Actualmente, son numerosos los trabajos que se realizan y se publican, las aplicaciones nuevas que surgen y las empresas que lanzan al mercado productos nuevos, tanto hardware como software, sobre todo para simulación.

### Definiciones de una red neuronal

Existen numerosas formas de definir a las redes neuronales; desde las definiciones cortas y genéricas hasta las que intentan explicar más detalladamente qué son las redes neuronales. Por ejemplo:

- 1) Una nueva forma de computación, inspirada en modelos biológicos.
- 2) Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.
- 3) Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.
- 4) Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.”[5]

Otra definición de redes neuronales podría verse en el siguiente párrafo:

“Las redes de neuronas artificiales (denominadas habitualmente como RNA o en inglés como: "ANN") son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales.”[4]

Como vemos hay múltiples definiciones de redes neuronales y múltiples formas de expresar qué es exactamente una red neuronal. No obstante la definición en sí no nos dice nada, sino más bien las características y el funcionamiento de dichas redes.

Antes de explicar vamos a mostrar unos ejemplos de redes neuronales para tener la visualización de éstas y hacernos una idea de todo lo que vamos a hablar más tarde.



## Eloise, Red Neuronal

---

Debemos tener en cuenta siempre que una red neuronal por tanto está basada en neuronas y en conexiones sinápticas. Y que se encargan de transmitir impulsos nerviosos para generar un conjunto de salidas a partir de un conjunto de entradas.

Para visualizar estas redes neuronales se suele utilizar su estructura, que quedará fácilmente representado por un grafo dirigido cuyos nodos serían las neuronas. Algunas representaciones podrían ser:

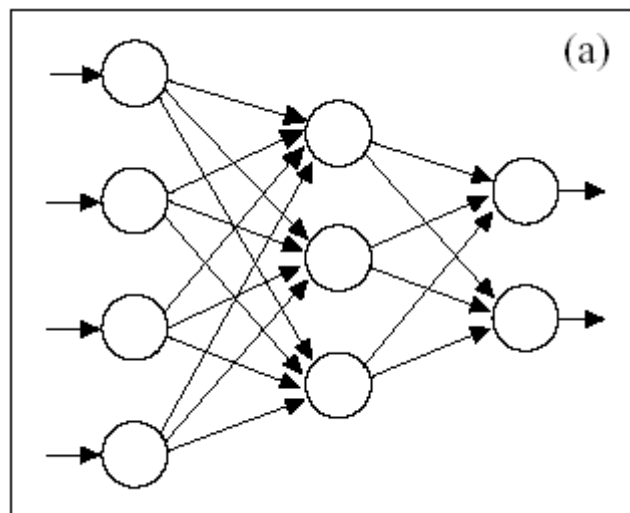


Imagen 1. Ejemplo a de Estructura de Red Neuronal.

En este caso nos encontraríamos en el caso de una red multicapa hacia adelante pues la salida de cada unidad siempre va hacia adelante.

Los nodos representan la unidad o neurona artificial y los arcos representan por tanto las uniones sinápticas que existen entre ellas.



## Eloise, Red Neuronal

---

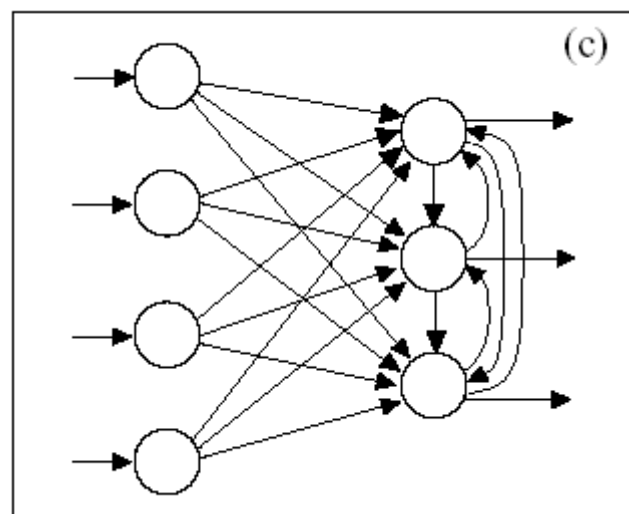
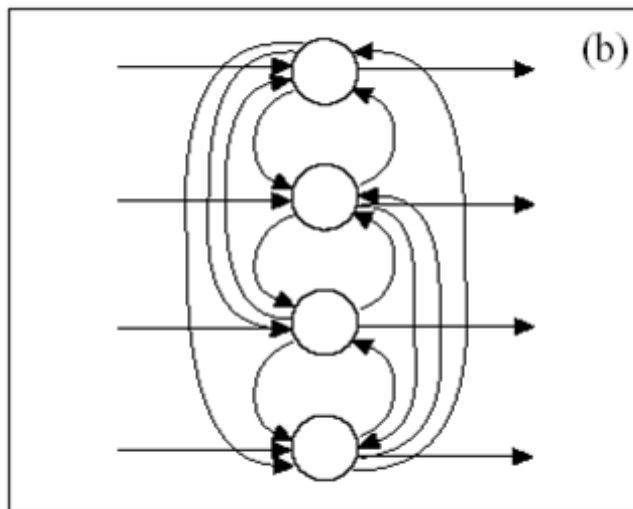


Imagen 2. Ejemplo b de Estructura de Red Neuronal.

Imagen 3. Ejemplo c de Estructura de Red Neuronal.

Pero como podemos observar no son ni mucho menos el único tipo de redes neuronales y las conexiones sinápticas pueden ser retro-alimentarias (Ejemplo b) o ir a otras neuronas de otras capas o de la misma capa de la red, independientemente de cual sea.



## Eloise, Red Neuronal

Nosotros trabajaremos, no obstante, solamente con las redes multicapa hacia adelante.

### Características de una Red Neuronal

Una red neuronal está constituida por nodos y arcos. Cada nodo representa a una neurona o unidad artificial. Los arcos dirigidos representan por tanto la conexión axón-dendritas.

Cada arco se dirige desde una unidad  $i$  hasta otra unidad  $j$ . Propaga una salida de la neurona  $i$  denominada  $a_i$ , ésta servirá como entrada de la siguiente neurona, la unidad  $j$ . Además, cada arco tiene asociado un peso numérico  $w_{ij}$  que determina la fuerza y el signo de la conexión, para así poder simular la sinapsis. [2]

Unidad Neuronal:

Posee un conjunto de entradas que servirán para calcular su salida. Además poseerá una entrada ficticia  $a_0 = -1$ . Con un peso umbral llamado  $w_{0i}$  (umbral de la unidad  $i$ ) que determinará la fuerza que deberán superar el conjunto de todas las entradas de la neurona (de ahí que  $a_0$  sea  $-1$ ).

Cada neurona además tendrá una función  $g$ , de activación, que actuará una vez superado el umbral para obtener la salida  $a_i$ , de la unidad. En una red neuronal, esta función será la misma para todas las neuronas. Permitiendo, así, el aprendizaje automático.

Una unidad por tanto será algo tal que así:

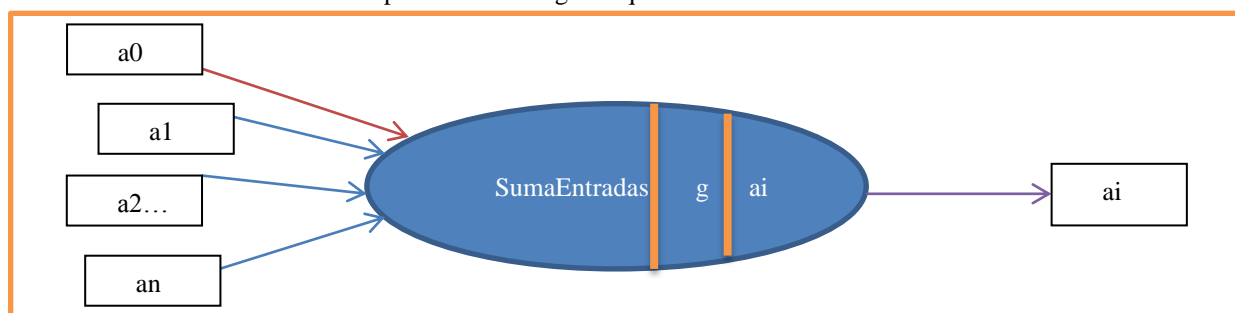


Imagen 4. Neurona Artificial.

Las funciones de activación más usadas son la umbral, la bipolar y la sigmoide. No procederemos a explicar nada más que la sigmoide pues las otras dos, Eloise, el proyecto, no las tendrá en cuenta.



## Eloise, Red Neuronal

---

La función de activación sigmoide es la siguiente:  
 $\text{Sigmoide}(x) = 1 / (1 + \text{exponencial}^{\wedge}(-x))$ .  
Cuya derivada es:  
 $\text{Sigmoide}'(x) = \text{Sigmoide}(x) * (1 - \text{Sigmoide}(x))$ .

Las redes neuronales multicapa hacia adelante se basan en 3 tipos de capa:

1. Capa de Entrada: 1 sola capa que reúne a las unidades de entrada, es decir todas las entradas que suministramos nosotros a Eloise para que ella, haga su trabajo y devuelva las salidas que correspondan.
2. Capas Ocultas: Un determinado número de capas que quedan ocultas en la red neuronal debido a que no sabemos que es lo que pasa y ocurre dentro de ellas, sirven para propagar las salidas y ayudar al aprendizaje de la red.
3. Capa de Salida: 1 última capa en la que se situarán las neuronas de salida, las cuales nos devolverán nuestras salidas.

Con una sola capa oculta es suficiente para la mayoría de las aplicaciones reales pero a medida que aumentamos la estructura, aumentaremos la capacidad expresiva de nuestra red.

Por tanto las redes neuronales multicapa hacia adelante funcionan por capas:

1. Suministramos las entradas numéricas a la capa de entrada.
2. Estas se propagan según el umbral y la función de activación a través de las capas ocultas (si las hay), y al final llegan a la capa de salida.
3. La capa de salida nos devuelve el conjunto de salidas que la red neuronal nos ha proporcionado.

Generalmente el aprendizaje se realiza con un conjunto de entrenamiento conocido es decir, conocemos las salidas reales de cada conjunto de entradas y por lo tanto, la red aprendería cambiando sus pesos correspondientes hasta obtener las salidas que más se asimilen a las reales.

Una vez la red esté entrenada (aprendizaje completo) podremos usarla para clasificar un objeto (por ejemplo) que no esté clasificado pasando las entradas correspondientes a ese objeto, que serán valores numéricos correspondientes a sus características.



## Eloise, Red Neuronal

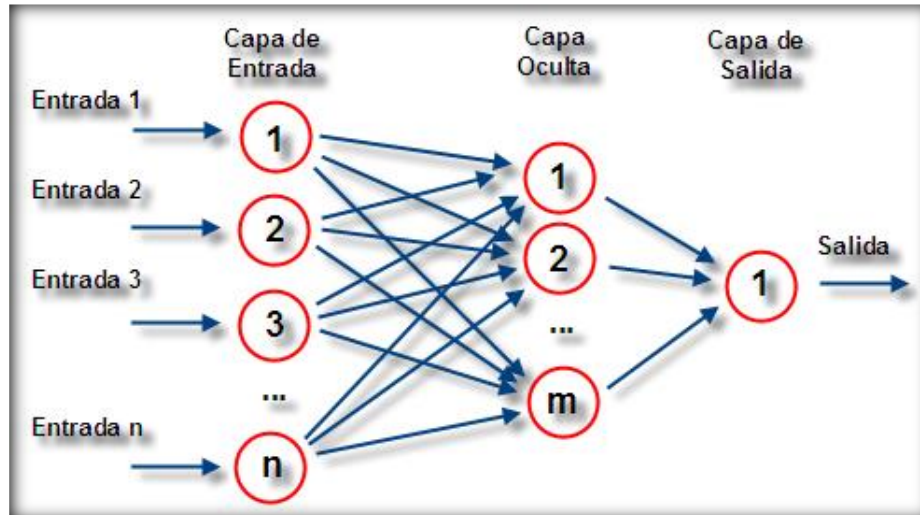


Imagen 5. Red Multicapa hacia adelante de una capa oculta.

Debemos tener en cuenta que además de éstas hay más tipos de redes neuronales como pueden ser:

- Redes Monocapas: 1 neurona por capa.
- Redes Retroalimentadas: Salidas vuelven a la entrada de la red.
- Perceptrones: una sola unidad de salida sin capas ocultas forma la red.
- Otras especiales: Caso de estructuras especiales como Adaline o Hopfield.



## Eloise, Red Neuronal

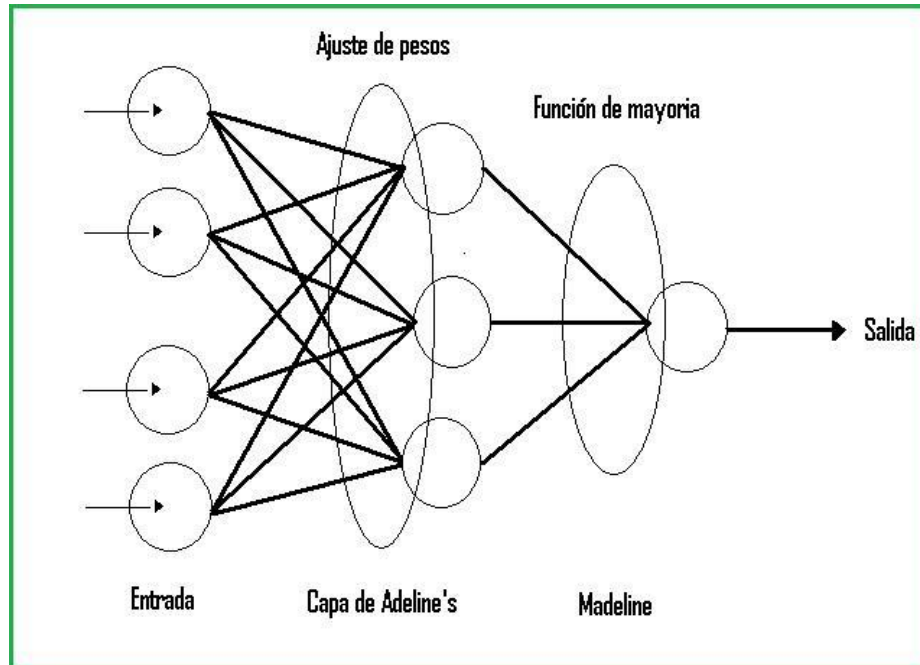


Imagen 6. Red Adaline.

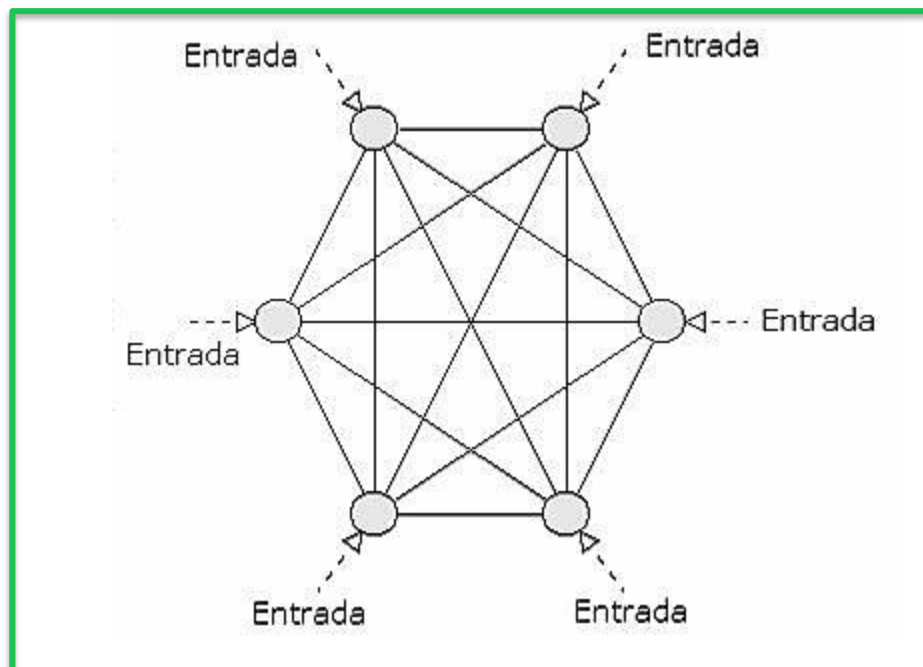


Imagen 7. Red Hopfield.



## Eloise, Red Neuronal

---

Por último en este apartado vamos a destacar algunas de las ventajas de usar redes neuronales.

### Ventajas:

“Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

1. Aprendizaje Adaptativo: capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
2. Auto-organización: una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje. (Clasificación).
3. Tolerancia a fallos: la destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
4. Operación en tiempo real: los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
5. Fácil inserción dentro de la tecnología existente: se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

### Aprendizaje adaptativo

La capacidad de aprendizaje adaptativo es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni necesidad de especificar funciones de distribución de probabilidad.

Las redes neuronales son sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.





## Eloise, Red Neuronal

---

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado su período de entrenamiento.

La función del diseñador es únicamente la obtención de la arquitectura apropiada. No es problema del diseñador el cómo la red aprenderá a discriminar. Sin embargo, sí es necesario que desarrolle un buen algoritmo de aprendizaje que le proporcione a la red la capacidad de discriminar, mediante un entrenamiento con patrones.

### Auto-organización

Las redes neuronales emplean su capacidad de aprendizaje adaptativo para autoorganizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es la modificación de cada elemento procesal, la autoorganización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas autoorganizan la información usada. Por ejemplo, la red llamada backpropagation, creará su propia representación característica, mediante la cual puede reconocer ciertos patrones.

Esta autoorganización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a las que no había sido expuesta anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tiene que solucionar problemas en los cuales la información de entrada no es muy clara; además permite que el sistema dé una solución, incluso cuando la información de entrada está especificada de forma incompleta.

### Tolerancia a fallos

Las redes neuronales fueron los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad cuando sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un número no muy grande de neuronas y aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina.

Hay dos aspectos distintos respecto a la tolerancia a fallos:



## Eloise, Red Neuronal

---

- a) Las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos. Esta es una tolerancia a fallos respecto a los datos.
- b) Las redes pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

La razón por la que las redes neuronales son tolerantes a los fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizado y direccionable. En cambio, las redes neuronales almacenan información no localizada. Por lo tanto, la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.

### Operación en tiempo real

Una de las mayores prioridades, casi en la totalidad de las áreas de aplicación, es la necesidad de realizar procesos con datos de forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínimo.

### Fácil inserción dentro de la tecnología existente

Una red individual puede ser entrenada para desarrollar una única y bien definida tarea (tareas complejas, que hagan múltiples selecciones de patrones, requerirán sistemas de redes interconectadas). Con las herramientas computacionales existentes (no del tipo PC), una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo coste. Por lo tanto, no se presentan dificultades para la inserción de redes neuronales en aplicaciones específicas, por ejemplo de control, dentro de los sistemas existentes. De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas en forma incremental y cada paso puede ser evaluado antes de acometer un desarrollo más amplio.

Redes neuronales y computadoras digitales.

Para entender el potencial de la computación neuronal, sería necesario hacer una breve distinción entre sistemas de computación neuronales y digitales: los sistemas neurológicos no aplican principios de circuitos lógicos o digitales.

Un sistema de computación digital debe ser síncrono o asíncrono. Si fuera asíncrono, la duración de los impulsos neuronales debería ser variable para mantener uno de los



## Eloise, Red Neuronal

---

valores binarios por periodos de tiempo indefinido, lo cual no es el caso. Si el principio fuera síncrono, se necesitaría un reloj global o maestro con el cual los pulsos estén sincronizados. Éste tampoco es el caso. Las neuronas no pueden ser circuitos de umbral lógico, porque hay miles de entradas variables en la mayoría de las neuronas y el umbral es variable con el tiempo, siendo afectado por la estimulación, atenuación, etc. La precisión y estabilidad de tales circuitos no es suficiente para definir ninguna función booleana. Los procesos colectivos que son importantes en computación neuronal no pueden implementarse por computación digital. Por todo ello, el cerebro debe ser un computador analógico.

Ni las neuronas ni las sinapsis son elementos de *memoria biestable*. Todos los hechos fisiológicos hablan a favor de las acciones de las neuronas como integradores analógicos, y la eficiencia de la sinapsis cambia de forma gradual, lo cual no es característico de sistemas biestables.

Los circuitos del cerebro no implementan *computación recursiva* y por lo tanto no son *algorítmicos*. Debido a los problemas de estabilidad, los circuitos neuronales no son suficientemente estables para definiciones recursivas de funciones como en computación digital. Un algoritmo, por definición, define una función recursiva.”[5]

Además de los aspectos dichos anteriormente posee otras ventajas como pueden ser utilizadas para generar sistemas dinámicos si usamos redes retro-alimentadas o también llamadas recurrentes.

### 1.2 PROBEN1

PROBEN1 es una colección de problemas que sirve para el aprendizaje de redes neuronales, y que contiene además una serie de reglas que todos sus problemas deben seguir. Contiene 15 conjuntos de datos pertenecientes a 12 campos distintos. Todos ellos representan problemas reales, excepto uno. Se presentan en el mismo formato. Las reglas mencionadas anteriormente detallan la forma de documentar los resultados tras unos algoritmos de dichos conjuntos.

PROBEN1 nace de la necesidad de ejemplos reales para entrenar redes neuronales y de la dificultad de conseguirlos. Con ello podemos evaluar los algoritmos de una forma concluyente y estandarizada.

Las reglas servirán para determinar cómo de bueno es un algoritmo con respecto a otros en función de su aprendizaje y rapidez, y por ello es necesario que el resultado de la evaluación quede guardado en tablas comparativas. Estas reglas por tanto determinarán que pruebas han de realizarse, y que resultados han de mostrarse para los conjuntos de datos.



## Eloise, Red Neuronal

---

### Ámbito de uso:

PROBEN1 sólo es válido para aprendizaje supervisado ya que requiere de un conjunto de entradas y otro conjunto salidas separados (no hay retroalimentación). Los problemas son todos independientes unos de otros. Los datos son todos numéricos (Reales o Booleanos binarios). Todos los problemas son reales, salvo el de las setas.

### Reglas:

Como generalidad tenemos que buscar:

- Validez.
- Reproducibilidad: debe poderse volver a experimentar.
- Comparabilidad: deben poderse comparar resultados.

Durante las pruebas:

- Conjunto de entrenamiento: para entrenar la red neuronal. Es el aprendizaje de la red.
- Conjunto de validación: para comprobar si la red ya ha aprendido bastante. Sino, deberemos volver a entrenarla desde donde está.
- Conjunto de testeo: una vez la red esté validada, será válida para testear los resultados de los datos.

Hay que declarar siempre los ejemplos que son de cada tipo (entrenamiento, validación y testeo). En PROBEN1 todos los conjuntos de datos tienen un 50% de entrenamiento y un 25% de validación y de testeo.

Por tanto el entrenamiento consiste en entrenar el conjunto de entrenamiento + validar el conjunto de validación. Una vez superada la validación se pasaría al test. Este tipo de procedimiento se denomina validación cruzada.

Entradas y salidas: dependerán del tipo de problema y podrán ser representadas mediante:

- ❖ Valores Reales: usualmente escalados entre un rango.
- ❖ Valores Enteros: son tratados como reales.
- ❖ Valores Ordinales: cómo el código binario.
- ❖ Valores Nominales: valores enteros para nombres.
- ❖ Valores Perdidos: 1 si el valor está perdido.

El Algoritmo de entrenamiento por tanto se basa en:

1. Elegir los parámetros de clasificación. Y ponerlos en formato PROBEN1
2. Calcular los errores de la siguiente forma:

Porcentaje del error =

$$100 * ((\text{SalidaMaxima} - \text{SalidaMinima}) / \text{N}^{\circ} \text{neuronasSalida} * \text{N}^{\circ} \text{Ejemplos}) * \text{Suma}$$



## Eloise, Red Neuronal

---

Donde Suma es la Suma por cada ejemplo de la Suma de todos los errores cuadráticos de salida de ese ejemplo es decir  $(SalidaEsperada - salidaVerdadera)^2$ .

3. Entrenar la red con respecto a los errores.
4. Validar la red.
5. Si no se valida volver a entrenarla.

Topologías de la red.

Deberemos además definir las estructuras de la red de la siguiente forma:

Número-ListaNúmeros-Número

Donde el primero indica el número de entradas luego cada número de la lista las neuronas ocultas de cada capa oculta (tantas como números haya en la lista) y por último el número de neurona de la capa de salida.

Además hay que especificar el rango de pesos, y la función de activación (en los nodos de entrada la función de activación siempre será la identidad).

### Resultados del Entrenamiento:

Entrenar la red no nos muestra resultados para el informe, sino más bien prepara la red para que los resultados que muestre sean concluyentes. Para ello es necesario entrenar la red con una serie de valores aleatorios, ya que debemos ir probando cosas distintas hasta validar la red.

Los errores de la red nos ayudarán a saber cuándo la red está lista. A veces es necesario pasar por alto algún valor outlier (valor bastante alejado de lo esperado), pero deben ser menos de un 10%.

Para entrenar el conjunto de entrenamiento hemos de decidir cuántas veces vamos a procesar cada ejemplo (esto se conoce como época). Nosotros usaremos como número de épocas el 5. Así cada ejemplo será procesado 5 veces en cada iteración.

Los pesos iniciales de la red serán aleatorios entre -1 y 1. Al menos los que nosotros usaremos. Normalmente suele usarse otros pesos  $(-1/N^{\circ}\text{neuronas entradas} \dots 1/N^{\circ}\text{neuronas entradas})$  o  $(-0.1 \dots 0.1)$  no obstante por facilidad nosotros usaremos el mencionado anteriormente.

### Problemas:

1. Cáncer: clasificar si un tumor es benigno o maligno. (9 entradas, 699 ejemplos).



## Eloise, Red Neuronal

---

2. Tarjeta de Crédito: predecir si será aceptada o no (51 entradas, 690 ejemplos).
3. Diabetes: si una persona tiene o no diabetes (8 entradas, 768 ejemplos).
4. Genes: decidir si la unión de dos nucleótidos es donadora o de aceptación (120 entradas, 3175 ejemplos)
5. Cristal: clasificar el cristal en 6 tipos. (9 entradas, 214 ejemplos).
6. Corazón: predecir enfermedad de corazón (35 entradas, 920 ejemplos). Otro conjunto de datos para este (heartc: 35 entradas, 303 ejemplos),
7. Caballos: predecir si un caballo vivirá, morirá o será sacrificado (58 entradas, 364 ejemplos).
8. Setas: venenosas o no (125 entradas, 8124 ejemplos).
9. Frijoles: clasificar de qué tipo de los 19 que hay es un frijol (35 entradas, 683 ejemplos).
10. Tiroides: si un paciente de tiroides esta en sobre-funcional, normal o bajo-funcional (21 entradas, 7200 ejemplos).
11. Edificación: predecir cuanta energía se consume en la edificación de electricidad, agua fría y agua caliente (14 entradas, 4208 ejemplos).
12. Llamadas solares: predecir número de llamadas solares en 24 horas divididas en grandes, medianas y pequeñas (24 entradas, 1066 ejemplos).
13. Corazón: predecir el porcentaje de qué enfermedad puede tener un humano (35 entradas, 920 ejemplos) y heartac (35 entradas, 303 ejemplos).

Para realizarlo se usará el algoritmo RPROP que definiremos en la siguiente sección.

Además PROBEN1 viene con tablas de resultados de diferentes redes neuronales entrenadas sobre sus ejemplos para poder comparar estos y ver cómo debe ser el entrenamiento de una red. Nosotros entrenaremos una con dichos ejemplos.

Por último queda por definir el formato de los archivos de PROBEN1.

PROBEN1 se estructura en carpetas por conjuntos de datos y en cada carpeta está el nombre del problema a tratar. Dentro hay un “readme” y además varios archivos .dt de datos con el siguiente formato:

```
bool_in=0
real_in=9
bool_out=6
real_out=0
training_examples=107
validation_examples=54
test_examples=53
0.281387 0.36391 0.804009 0.23676 0.643527 0.0917874 0.261152 0 0 1 0 0 0 0
0.260755 0.341353 0.772829 0.46729 0.545966 0.10628 0.255576 0 0 0 1 0 0 0
[urther data lines deleted]
```



## Eloise, Red Neuronal

---

Las primeras siete líneas determinan las entradas y salidas booleanas y reales, el conjunto de entrenamiento, el conjunto de validación y el conjunto de test. Después de esto vienen todos los ejemplos (1 por línea, con sus valores separados por espacios en blanco). Primero las entradas, y al final las salidas de cada ejemplo en cada línea. Por tanto ya queda explicado PROBEN1, y cómo está realizado, así pues podremos utilizarlo sin tener que detenernos a explicarlo más veces.

## 2 Algoritmos

En esta sección explicaremos como funciona Eloise.

Para empezar Eloise es un proyecto implementado en Java, concretamente en Eclipse, con interfaz gráfica diseñada en NetBeans y datos usados del PROBEN1.

Partimos del punto de la necesidad de crear redes neuronales para ello, Eloise posee varias clases:

- Neurona: Objeto Abstracto.
- NeuronaEntrada: Son las unidades de entrada.
- NeuronaSalida: Son las unidades de salida con su uniones y pesos.
- NeuronaOculta: Son las unidades de las capas ocultas.
- Arco: Representa una unión dirigida con un peso.

Las anteriores definen por tanto la clase Red Neuronal:

- Red Neuronal: Cuyo constructor viene dado por Integer, List<Integer>, Integer definiendo el número de neuronas de entrada, capas ocultas y salida respectivamente.

Clases de utilidad:

- GeneradorID: genera Ids para las Neuronas.
- Propagacion Hacia Atrás: Ayudará a la retropropagación
- Propagacion Adelante: Ayudará a la retropropagacion y a la validación cruza.

Clases de uso:

- Metodos: Posee los algoritmos y funciones necesarias para usar la red.
- Test: Prueba las redes neuronales y los ejemplos pasándole la ruta del archivo.

Así queda dicho que la clase java se puede entender fácilmente leyendo, con paciencia el código.



## Eloise, Red Neuronal

---

Ahora es el momento de explicar los algoritmos en los que se basa Eloise para reconocer y clasificar los ejemplos. Se basa en 3 algoritmos:

1. Algoritmo de Propagación hacia adelante.
2. Algoritmo de Retropropagación.
3. Algoritmo de Validación cruzada.

1. Algoritmo de Propagación hacia adelante:

Partimos de un conjunto de entradas y una red neuronal.

Las unidades propagan su salida usando su función de activación a la suma de todas sus entradas validadas según sus pesos (multiplicación). Y dichas salidas, se van propagando de tal manera que la unidad de una capa otorgará su salida como entrada a todas las unidades de la siguiente capa hasta llegar a la de salida que será la salida final del algoritmo.

2. Algoritmo de Retropropagación:

Partimos de un conjunto de entradas, una red neuronal, una constante llamada factor de Aprendizaje, un conjunto de salidas esperadas, y una constante más llamada momentum.

- I. Se propaga las entradas hacia adelante hasta obtener las salidas mediante el algoritmo de Propagación hacia adelante.
- II. Se calcula el error observando la salida obtenida y la salida esperada.
- III. Se calcula cuánto supone ese error gracias a los pesos de las unidades y sus arcos.
- IV. Se modifican los pesos para hacer que la salida sea igual a la salida esperada. Teniendo en cuenta un factor de aprendizaje que indica cuando aprenderá la red cada vez que se ejecute este algoritmo (Esto se hace pues si aprende siempre el 100% la red no terminará de aprender algo para un conjunto de ejemplos ya que cambiarán sus pesos todo el rato en cada ejemplo distinto sin llegar a un punto medio), además se tiene en cuenta el momentum, que es una constante que se multiplica por todas las veces que se ha modificado el valor de cada peso y se tiene en cuenta las actualizaciones en cada momento.
- V. Así la red irá aprendiendo cada vez que se ejecute este algoritmo.
- VI. El algoritmo se ejecutará para cada ejemplo que tengamos de un conjunto de entrenamiento dado.





## Eloise, Red Neuronal

---

### 3. Algoritmo de validación cruzada:

Dicho algoritmo se utiliza para saber cuándo una red está lo suficientemente entrenada.

Para ello se sigue el siguiente esquema:

- I. Partimos de un conjunto de entrenamiento (entrenamiento, validación, test), una red, un momentum y lo necesario para los anteriores algoritmos.
- II. Hasta que se cumpla cierta o ciertas condiciones:
  - II.I Para cada ejemplo de entrenamiento del conjunto de entrenamiento realizaremos el algoritmo de retropropagación.
  - II.II Repetiremos II.I un número  $k$  de veces (Épocas).
  - II.III Terminadas las  $k$  épocas pasaremos a una fase de validación en donde se usará una propagación hacia adelante y se comprobarán errores.
  - II.IV Volver al paso II. (Esto es una iteración).
- III. Una vez las condiciones estén cumplidas, pasamos a realizar el test y a mostrar los resultados.

Eloise usa este algoritmo con 3 condiciones de parada y  $k = 5$ :

- 3000 Iteraciones.
- $GL > 5$ , donde usaremos  $GL$  con 5 épocas y es el error mínimo de validación.
- $Pk < 0.1$ , donde usaremos  $k=5$  y es el error de minimización del error a menos del 10%.

Los errores pueden verse en el PROBEN1 que están descritas sus fórmulas [1].

Eloise te muestra una serie de datos relevantes y comparables para un problema dado. Estos datos son las tablas que se mostrarán en la siguiente sección.

Eloise guarda los datos en un fichero.txt (el cual se sobrescribe cada vez que se ejecuta) y realiza las gráficas correspondientes para poder visualizarlos fácilmente, que guarda en la carpeta imágenes, y qué también sobrescribe.

Por último nuestro algoritmo hace uso de las funciones: media y desviación típica para mostrar dichos resultados, además de que debe usar variables específicas para mostrarlos.

Para usar esto hemos creado el método rPROP que acepta un fichero, una List<Integer> para las capas ocultas y las constantes. (Por defecto 0 ocultas, 0,1, 0.9).



## Eloise, Red Neuronal

---

### 3 Experimentación

En esta sección pasaremos a ofrecer resultados obtenidos mediante Eloise.

Para que esta fase sea concluyente es preciso realizar más de un problema, por lo que hemos seleccionado tres.

Dichos problemas son: Cancer, Tiroides, Edificación.

#### 3.1 Descripción de los experimentos

Para cada problema seleccionado, Eloise ha realizado los siguientes experimentos:

1. Ejecución de cada uno de los 3 conjuntos de datos de cada problema en:
  - Red Neuronal con 0 capas ocultas.
  - Red Neuronal con 1 capa oculta con 2 Neuronas.
2. 5 Ejecuciones de cada uno de los mencionados anteriormente.
3. Pruebas de ejecución aleatoria con más capas ocultas y más neuronas, sin embargo estos datos no han sido guardados.

Los problemas escogidos son los siguientes:

1. Cáncer: clasificar si un tumor es benigno o maligno. (9 entradas, 699 ejemplos).
2. Tiroides: si un paciente de tiroides esta en sobre-funcional, normal o bajo-funcional (21 entradas, 7200 ejemplos).
3. Edificación: predecir cuanta energía se consume en la edificación de electricidad, agua fría y agua caliente (14 entradas, 4208 ejemplos).

Por tanto para el primero tendremos 2 salidas, para el segundo 3 y para el tercero otras 3.

Los resultados que se muestran son aquellos que se usan en el PROBEN1. Se mostrarán mediante tablas. Las arquitecturas de las redes no las mostraremos sino que realizaremos una tabla por arquitectura, pues es más fácil de ver.

Así pues en la siguiente sección se mostrarán por fin los resultados de nuestra pequeña Eloise.



## Eloise, Red Neuronal

### 3.2 Resultados obtenidos

#### CANCER

Resultados obtenidos:

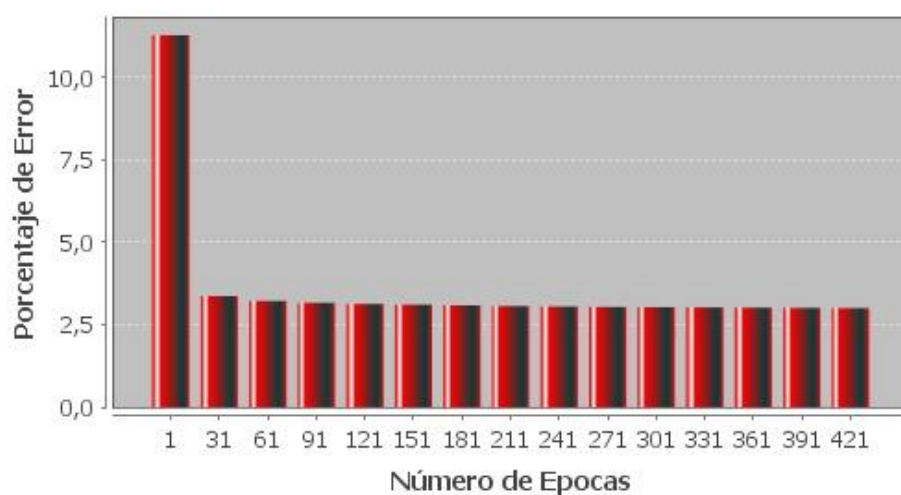
Red Neuronal sin capas ocultas:

Fichero		Training Set		Validation Set		Test Set		P	
		Media	DesvT	Media	DesvT	Media	DesvT		
Cancer1		3.17	0.63	2.03	0.29	0.82	5.08	0.40	
Cancer2		2.49	0.90	1.88	0.30	2.31	13.33	1.22	
Cancer3		2.03	0.69	3.12	0.30	0.96	4.57	0.31	
Fichero		Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
		Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Cancer1		2.87	16.73	0.28	0.23	213.0	122.83	62.5	35.36
Cancer2		4.02	19.67	1.58	1.80	98.0	56.43	37.5	20.92
Cancer3		4.60	20.97	1.69	1.77	145.5	83.86	50.0	28.13

Tabla 1. Cáncer sin capas ocultas.

A continuación mostramos las gráficas que Eloise nos genera para uno de estos ficheros: Cancer1.

### Errores De Entrenamiento

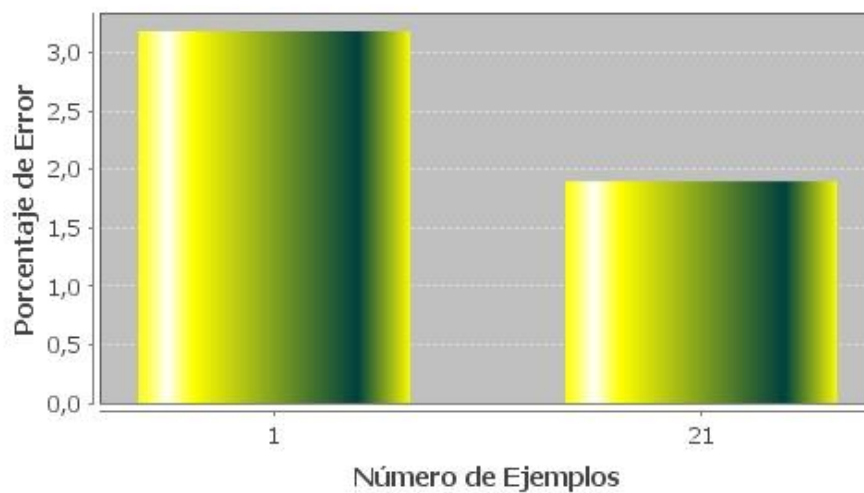




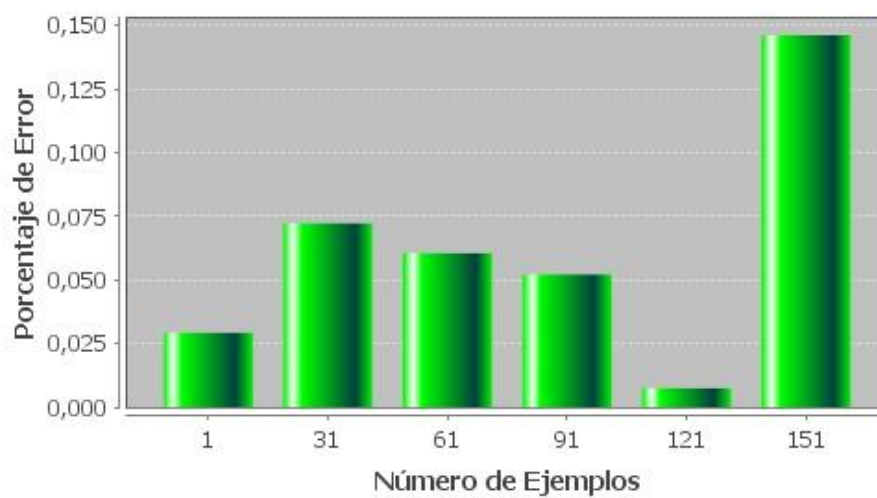
## Eloise, Red Neuronal

---

### Errores De Validacion



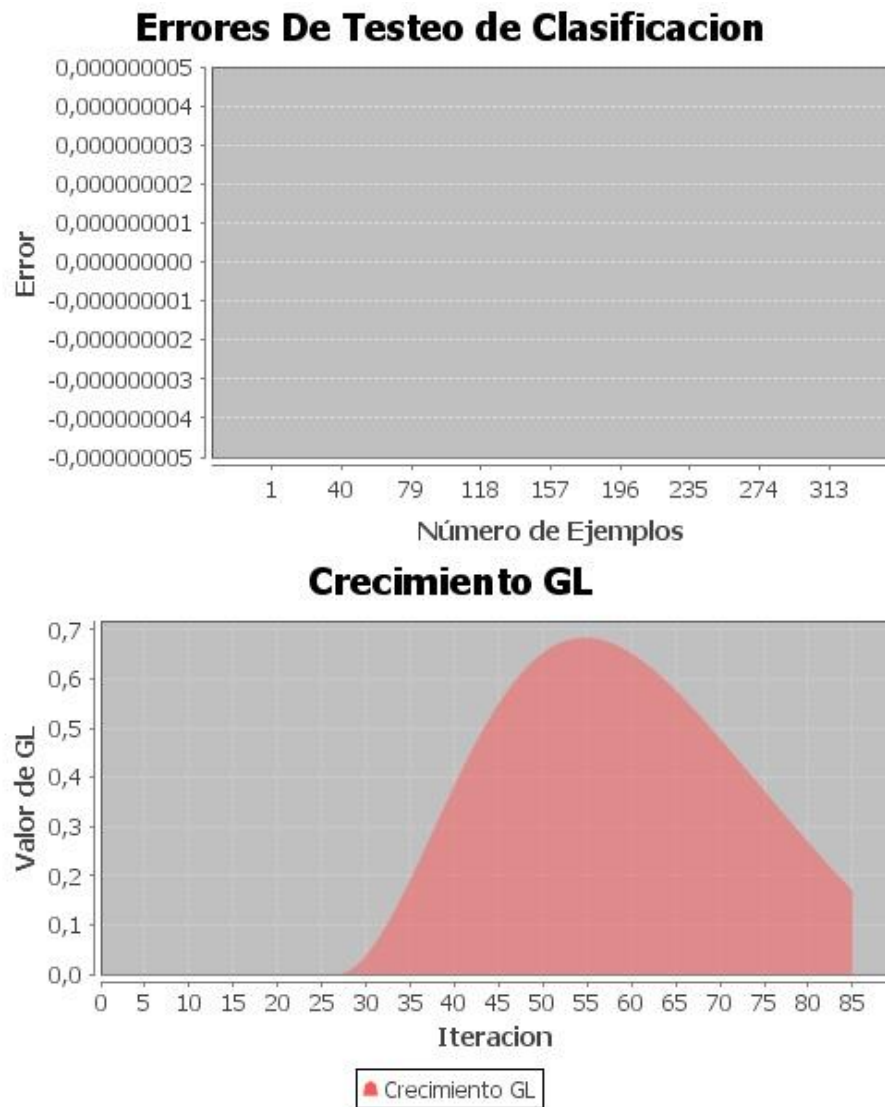
### Errores De Testeo





## Eloise, Red Neuronal

---



Imágenes 8-12. Cancer1.txt



## Eloise, Red Neuronal

Red Neuronal con capas ocultas (1):

Fichero		Training Set		Validation Set		Test Set		P	
		Media	DesvT	Media	DesvT	Media	DesvT		
Cancer1		3.19	0.90	1.97	0.41	0.38	1.04	0.20	
Cancer2		2.45	1.19	1.93	0.37	2.37	12.88	1.23	
Cancer3		2.54	1.86	3.15	0.45	1.14	5.50	0.36	
Fichero		Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
		Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Cancer1		1.72	13.04	1.90	1.83	78.0	44.89	25.0	13.69
Cancer2		4.02	19.67	1.96	1.98	43.0	24.68	17.5	9.35
Cancer3		4.59	20.97	1.81	2.19	28.0	16.02	12.5	6.45

Tabla 2. Cáncer con capas ocultas.

Y mostramos de nuevo las gráficas:

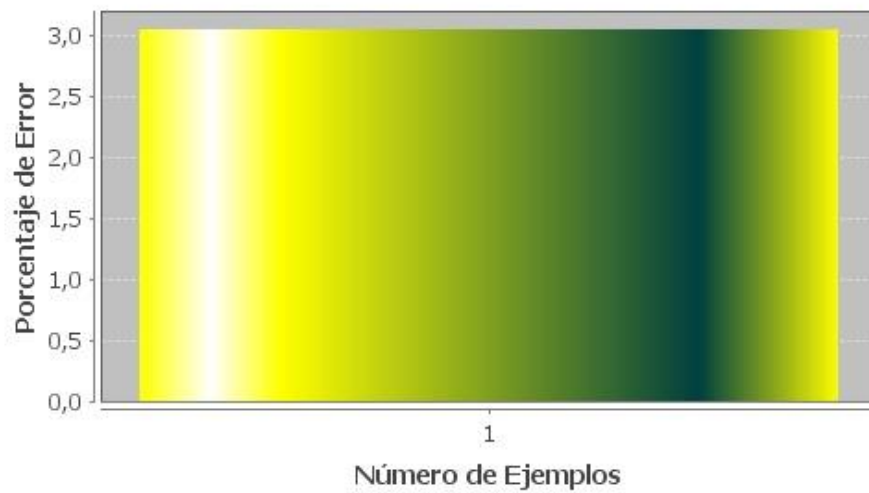




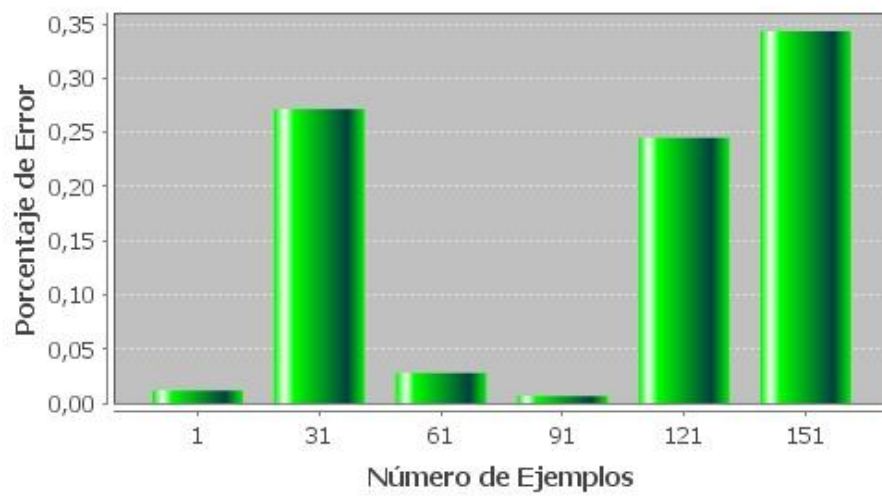
## Eloise, Red Neuronal

---

### Errores De Validacion



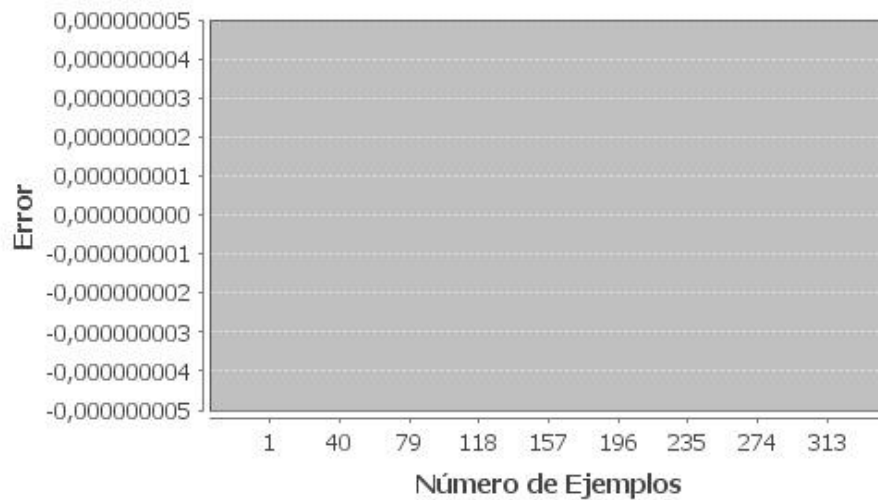
### Errores De Testeo



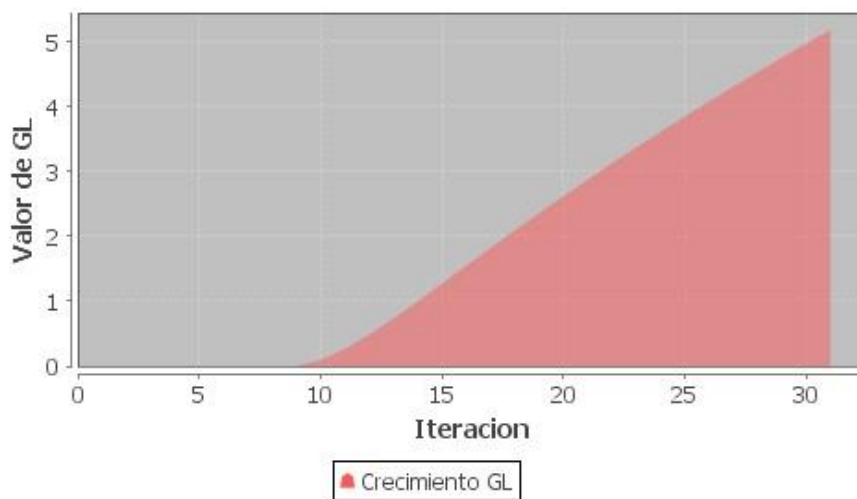


## Eloise, Red Neuronal

### Errores De Testeo de Clasificacion



### Crecimiento GL



Imágenes 13-17. Cancer 1. 1 capa oculta.

Ahora vamos a ver otro de los problemas.





## Eloise, Red Neuronal

### TIROIDES

Resultados obtenidos:

Red Neuronal sin capas ocultas:

Fichero		Training Set		Validation Set		Test Set		P	
		Media	DesvT	Media	DesvT	Media	DesvT		
Thyroid1		3.17	0.28	3.55	0.16	2.23	8.51	0.63	
Thyroid2		3.43	0.23	3.08	0.22	2.34	8.78	0.76	
Thyroid3		3.40	0.24	2.88	0.24	2.62	9.35	0.91	
Fichero		Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
		Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Thyroid1		3.77	19.07	0.0	0.0	808	466.35	808	466.35
Thyroid2		3.59	18.61	0.0	0.0	860	496.67	860	496.70
Thyroid3		3.96	19.51	0.0	0.0	970	560.17	970	560.17

Tabla 3. Tiroides sin capas ocultas.

Red Neuronal con capas ocultas (1):

Fichero	Training Set		Validation Set		Test Set		P	
	Media	DesvT	Media	DesvT	Media	DesvT		
Thyroid1	2.49	0.59	2.88	0.71	2.64	13.09	0.92	
Thyroid2	2.93	0.59	2.47	0.64	2.45	10.40	1.00	
Thyroid3	2.38	0.71	2.06	0.62	3.65	19.47	1.77	
Fichero	Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
	Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Thyroid1	2.61	15.95	0.42	1.02	318.0	183.45	304.5	150.20
Thyroid2	1.81	13.35	0.18	0.81	273.0	157.47	272.3	145.41
Thyroid3	2.44	15.44	0.30	0.88	545.5	314.80	452.35	246.36

Tabla 4. Tiroides con capas ocultas.

Las demás gráficas estarán guardadas en un archivo adjunto pues ocuparíamos demasiado. Carpeta: “Gráficas de los Ejemplos”.



## Eloise, Red Neuronal

### EDIFICACION

Resultados obtenidos:

Red Neuronal sin capas ocultas:

Fichero	Trainng Set		Validation Set		Test Set		P	
	Media	DesvT	Media	DesvT	Media	DesvT		
Building1	0.12	0.12	0.27	0.05	0.90	0.82	3.30	
Building2	0.33	0.30	0.25	0.01	0.79	1.00	3.12	
Building3	0.39	0.33	0.22	0.01	0.89	1.07	4.05	
Fichero	Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
	Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Building1	0.0	0.0	2.17	2.63	18.0	10.25	10.0	5.0
Building2	0.0	0.0	0.50	0.32	30.5	17.46	7.5	3.53
Building3	0.0	0.0	3.18	1.84	33.0	18.91	7.5	3.53

Tabla 5. Edificación sin capas ocultas.

Red Neuronal con capas ocultas (1):

Fichero		Training Set		Validation Set		Test Set		P	
		Media	DesvT	Media	DesvT	Media	DesvT		
Building1		0.06	0.02	0.24	0.11	0.82	0.80	0.92	
Building2		0.18	0.04	0.21	0.03	0.85	1.29	3.94	
Building3		0.19	0.03	0.19	0.03	0.92	1.23	4.93	
Fichero		Test Set Clasification		Overfit		Epocas		Epocas Relevantes	
		Media	DesvT	Media	DesvT	Media	DesvT	Media	DesvT
Building1		0.0	0.0	0.10	0.19	331.6	196.72	293.0	169.02
Building2		0.0	0.0	0.10	0.12	45.5	26.12	25.0	13.69
Building3		0.0	0.0	0.88	0.82	68.0	39.12	27.5	15.14

Tabla 6. Edificación con capas ocultas.

Para finalizar esta sección hemos de decir que los datos suministrados en las tablas anteriores no servirán para cuantificar cuán



## Eloise, Red Neuronal

---

### 4 Conclusiones y trabajo futuro

Las redes neuronales tienen mucho potencial y la conclusión más clara que podemos sacar de este proyecto es que, podemos sacarle mucho provecho en nuestro futuro laboral.

Investigar en este ámbito computacional puede ser muy útil porque estamos trabajando en la clasificación de objetos, además de muchas otras aplicaciones y eso, es algo que siempre es ventajoso.

Por ello es importante tener redes neuronales, tener ejemplos para poder entrenarlas e investigar más profundamente en ello.

### Referencias bibliográficas

1. Lutz Prechelt, PROBEN1 – A set of Neural Network Benchmark Problems and Benchmarking Rules, Technical Report 21/94, 1994.
2. S. Russel, P. Norvig. Inteligencia Artificial: un enfoque moderno, Capítulo 20, Prentice Hall, 3ª ed. (2010).
3. Guía de autores para la serie Lecture Notes in Computer Science, de Springer  
<http://www.springer.com/librarians/e-content?SGWID=0-113-12-558799-0>
4. Wikipedia  
<http://es.wikipedia.org/wiki/Neurona>  
[http://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](http://es.wikipedia.org/wiki/Red_neuronal_artificial)
5. Goica  
<http://www.monografias.com/trabajos12/redneuro/redneuro.shtml#intro>