Clasificación de información usando Redes Neuronales

Enrique Alexis Peinado Rodriguez Ingrid Ipanaqué Casquina

***Resumen*—El cerebro humano es un sistema tan complejo que supera la capacidad de cualquier maquina, en el sentido de poder responder a nuevas situaciones, poder aprender de ellas y dar una respuesta adecuada a un cada una de ellas. Para ello existe un campo de la computación, llamado Computación Congnitiva que se encarga de desarrollar ordenadores que piensen de forma humana. Dentro del campo de estudio se tiene una variedad de metodologías que tratan de poder realizar este fin, entre ellas tenemos el concepto de Redes Neuronales el cual trata de imitar el comportamiento que tiene un neurona, utilizando esta metodología se utilizará para poder gestionar información no estructurada, utilizando el aprendizaje supervisado y no supervisado.**

I. *Análisis del Tema Investigado*

I-A. Red Neuronal Artificial

Una **red neuronal o una red neuronal artificial(RNA)** puede definirse como un sistema de procesamiento de información compuesto por un gran número de elementos de procesamiento (neuronas), profusamente conectados entre sí a través de canales de comunicación. Estas conexiones establecen una estructura jerárquica y permiten la interacción con los objetos del mundo real tratando de emular al sistema nervioso biológico. A continuación se muestra unos de los 5 principios más importantes del funcionamiento de las RNA:

- ***Aprendizaje adaptativo:***
  Las redes neunorales 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 constatemente cambiando para adaptarse a las nuevas condiciones.

  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 continuan aprendiendo a lo largo de su vida, después de completado su periodo se entrenamiento.

- ***Autoorganización:***
  La auto-organización consiste en la modificación de toda la red completa con el fin de llevar a cabo un objetivo específico. Autoorganziación significa generalización, de esta forma una red puede responder a datos o situaciones que no ha experimentado antes, pero que puede inferir en base a su entrenamiento. Esta característica es muy útil sobre todo cuando la información de entrada es poco clara o se encuentra incompleta.

- ***Tolerancia a fallos:***
  La tolerancia a fallos se entiende aquí en dos sentidos: primero, las redes pueden **reconocer patrones de información con ruido, distorsión o incompletos** (tolerancia de fallos respecto de los datos); y segundo, pueden **seguir trabajando (con cierta degradación)** aunque se destruya parte de la red (tolerancia a fallos respecto de la estructura).

  La explicación de este fenómeno se encuentra en que, mientras la computación tradicional almacena la información en espacios únicos, localizados y direccionables, las redes neuronales lo hacen de forma distribuida y con un alto grado de redundancia.

- ***Operación en tiempo real:***
  De todos los métodos existentes, la RNA son las más indicadas para el reconocimiento de patrones en tiempo real, debido a que **trabajan en paralelo** actualizando todas sus instancias simultáneamente.

- ***Fácil inserción en la tecnología existente:***
  Es relativamente sencillo obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilita la integración modular en los sistemas existentes.

I-A1. **Elementos de una RNA***:* Todas las RNA poseen elementos que las caracteriza y que además hace posible el entrenamiento de la red, las cuales son:

- **Elementos internos:** valores de entradas, señal de salida, peso de la sinapsis (factor asignado a cada sinapsis), entrada total, función de salida, función de activación y reglas de aprendizaje (permiten modificar los pesos de la sinapsis).

- **Capa o nivel:** conjunto de neuronas cuya capa tiene su origen en la misma fuente y cuyas salidas van al mismo destino.

- **Tipos de capas:** entrada (reciben estímulos externos), ocultas (representación interna de la información) y salida.

- **Conexión entre neuronas:** propagación hacia delante (ninguna salida de las neuronas es entrada del mismo nivel o niveles superiores) y propagación hacia detrás (la salida de las neuronas pueden ser entradas del mismo nivel o niveles anteriores y también de ellas mismas).

- **Dinámica:** asincrónica (evalúan su estado continuamente, según les llega información), sincronía (cambios a la vez en todas las neuronas).

I-B. Aprendizaje Automático:

Rama de la **Inteligencia Artificial** que se encarga de desarrollar nuevas técnicas que le permitirá a la computadora aprender, en otras palabras, de **crear programas capaces de generalizar comportamientos a partir de una informacion no estructurada**, es por tanto un proceso de inducción del concocimiento.

Además se dedica al estudio de los agentes/programas que aprenden o evolucionan basados en su experiencia, para realizar una tarea determinada cada vez mejor. El objetivo principal de todo proceso de aprendizaje es utilizar la evidencia conocida para poder crear una hipótesis y poder dar una respuesta a nuevas situaciones no conocidas.

Ante esto existen algoritmos para el aprendizaje automático, el cual se agrupan en una taxonomía en función de la entrada de los mismos, en este caso mencionaremos dos tipos:

- ***Aprendizaje supervisado:***
  El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que deberia generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que esta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada.

  Un ejemplo de este tipo de algoritmo es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar (clasificar) una serie de vectores utilizando una entre varias categorías (clases). La base de conocimiento del sistema está formada por ejemplos de etiquetados anteriores. En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo, que dan lugar a los siguientes aprendizajes supervisados:

  - **Aprendizaje por corrección de error:** Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red, es decir, en función del error cometido en la salida. Por ejemplo, la regla de aprendizaje del Perceptron.

  - **Aprendizaje por refuerzo:** Se trata de un aprendizaje supervisado, más lento que el anterior, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada.

  - **Aprendizaje estocástico:** Consiste basicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

  Este tipo de aprendizaje puede llegar a ser muy útil en problemas de investigación biológica, biología computacional y bioinformática.

- ***Aprendizaje no supervisado:***
  El proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos. Por lo tanto, en este caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas. Este es útil para la compresión de datos: fundamentalmente, todos los algoritmos de compresión dependen tanto explícita como implícitamente de una distribución de probabilidad sobre un conjunto de entrada.

  - **Aprendizaje hebbiano:** Pretende medir la familiaridad o extraer características de los datos de entrada. Este aprendizaje consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de activación (salidas) de las dos neuronas conectadas.

  - **Aprendizaje competitivo y comparativo:** Las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Se pretende que cuando se presente a la red cierta información, sólo una o un grupo de ellas se activen. El objetivo de este aprendizaje es categorizar (clustering) los datos que se introducen en la red.

Distincion entre Aprendizaje supervisado y no supervisado

El aprendizaje supervisado se caracteriza por contar con información que especifica qué conjuntos de datos son satisfactorios para el objetivo del aprendizaje. En el aprendizaje no supervisado el programa no cuenta con datos que definan que información es satisfactoria o no.

Criterio de clasificación por reglas de aprendizaje:

1. **ON-line:** El aprendizaje se efectúa durante el funcionamiento normal de la red, por lo que aquí no se diferencia la fase de aprendizaje de la de operación.

2. **OFF-line:** La red debe desconectarse de su funcionamiento hasta que el aprendizaje termine, para luego establecer de forma fija sus pesos y comenzar a operar.

La principal diferencia entre las dos estará en las redes off-line serán más estables en su funcionamiento que las on-line.

Enfoques

- Árboles de decisiones
- Reglas de asociación
- Algoritmos genéticos
- Redes neuronales artificiales
- Máquinas de vectores de soporte
- Máquinas de vectores de soporte
- Redes bayesianas

II. *Neurona y RNA:*

Como se dijo anteriormente las **RNA** tratan de imitar el comportamiento de una neurona, para ello se debe de tener en cuenta las características que posee una.

1. ***Neurona:***
   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.

   Características de una neurona:

   - **Dendritas:** Son las ramificaciones receptoras de la neurona, el cual carga señales electricas de exterior hacia el soma.
   - **Soma:** Es el cuerpo de la neurona es cual realiza la suma de las señales eléctricas proveniente de las dendritas, las procesa y envia la señal resultante hacia el axón.
   - **Axón:** Es una fibra aun más grande que las dendritas que se encargan de trasmitir la señal resultante del soma hacia otras neuronas.
   - **La sinapsis** es una unión intercelular que se da entre el axón de una neurona y las dendritas de otra. En estos contactos se lleva a cabo la transmisión del impulso nervioso.
   - **El impulso nervioso** es aquell señal eléctrica que se trasmite de neurona en neurona.

2. ***RNA***
   Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

   - **Entradas:** Conjunto de entradas las cuales pueden ser tanto discretas como continuas.
   - **Pesos Sinápticos:** Representa la intensidad de interacción entra la neurona presináptica y la neurona postsináptica.
   - **Función de propagación:** Proporciona el valor del potencial postsináptico de a neurona en funcion de sus pesos y entradas. Esta funciones dependiendo del método de aprendizaje podrían ser:

     a) Función de tipo lineal basada en la suma ponderada.
     $$h_{i(t)} = \sum_j^n W_{ij} X_j$$
     b) Función basada en la distancia euclediana.
     $$h_{i(t)} = \sqrt{\sum_j^n (X_j - W_{ij})^2}$$
   - **Función de activación:** Propociona el estado de activación actual de la neurona en función de su anterior y de su potencial postsináptico actual.
     $$a_{i(t)} = f_i(a_{i(t-1)}, h_{i(t)})$$
     Aunque en muchos modelos no depende de su estado anterior, entonces:
     $$a_{i(t)} = f_i(h_{i(t)})$$

Nombre	Función	Rango
Identidad	$y = x$	$[-\infty, +\infty]$
Escalon	$y = signo(x)$ Función Heaviside $y = h(x)$	$\{-1, +1\}$ $\{0, +1\}$
Lineal a tramos	$y = -1, x < -1$ $y = x, -1 \leq x \leq -1$ $y = -1, +1 \leq x$	$[-1, +1]$
Sigmoidea	$y = \frac{1}{1 + \exp -x}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$
Gaussiana	$y = A \exp -Bx^2$	$[0, +1]$
Sinusoidal	$y = Asen(wx + \varphi)$	$[-1, +1]$

III. *Limitaciones De Los Sistemas Tradicionales De Gestión De Documentación:*

Tradicionalmente, un profesional infiere una estructura a partir de información documental no estructurada que, implementada en una aplicación informática permite con posterioridad recuperar información. Esta estructuración se realiza habitualmente por el siguiente método:

- **El método de las tablas relacionales:** De la información no estructurada del documento se extraen una serie de palabras clave que lo identifican, para después recuperar esas palabras como valores de los campos de un registro predefinido. Lo que tenemos es, pues, un sistema que gestiona una referencia del documento.

Las principales limitaciones de estos métodos serían:

1. Los procesos de estructuración mediante estos métodos son subjetivos y rígidos.
2. Resulta poco operativo para el usuario frente a los autores de la aplicación.
3. Si con el paso del tiempo, hay nuevas necesidades de búsqueda, la tarea de reindexación es complicada, lenta y costosa.

IV. *Implementación:*

IV-A. Lenguaje R

El lenguaje R es un entorno y lenguaje de programación con un enfoque estadístico y además proporciona un amplio abanico de herramientas estadísticas(modelos lineales, modelos no lineales, test estadísticos, análisis de series temporales, algoritmos de clasficicación y agrupamiento) y gráficas.

Características:

- Es un lenguaje de programación el cual permite que lo usuarios creen sus propias funciones.
- Posee manipulación de objetos en R y además su **orientación a objetos.**
- La facil extensión de R debido a su política de **lexical scoping**
- La integración y la sencilla manipulación de base de datos.
- Su capacidad gráfica, permite generar gráficos de alta calidad.

IV-A1. Paquetes: Los paquetes son la unidades fundamentales de código R reproducible. Ellos incluyen funciones en R reutilizables, la documentacion que describe como usarlos, y datos de la muestra. A su vez R forma parte de un proyeto colaborativo y abierto, en el cual los usuarios pueden publicar paquetes que extienden su configuración básica.

La implementación se basará de acuerdo a dos paquetes los cuales implementan tanto algoritmos supervisados como no supervisado, los cuales son:

1. **neuralnet:**
   El paquete NeuralNet contiene una función muy flexible para entrenar las redes neuronales de alimentación directa, es decir, a la aproximación de una relación funcional. Puede manejar teóricamente un número arbitrario de covariables y variables de respuesta, así como de las capas ocultas y neuronas ocultas a pesar de que los costes computacionales pueden aumentar exponencialmente con el más alto grado de complejidad. Esto puede causar una parada temprana del proceso de iteración ya que el máximo de pasos de iteración, que pueden ser definidos por el usuario, se alcanza antes que converge el algoritmo. Además, el paquete proporciona funciones para visualizar los resultados o, en general, para facilitar el uso de las redes neuronales. Por ejemplo, el cálculo función se puede aplicar para calcular predicciones de nuevas combinaciones de covarianza.

2. **kohonen:**
   El paquete kohonen tiene como objetivo proporcionar funciones fáciles de usar para los mapas de auto-organización, con especial énfasis en la visualización.
   Las funciones básicas son:
   - **som**,la forma usual de mapas de auto-organización; **xyf**, para los mapas de auto-organización supervisados.
   - **x-y** para mapas fusionado, que son útiles cuando la información adicional se encuentran en forma de, por ejemplo, una clase de variable que está disponible para todos los objetos.
   - **bdk**, una formulación alternativa llamada mapas de Kohonen bidireccionales.
   - **la generalización de los mapas xyf** a más de dos capas de información, en la función supersom.

   Estas funciones pueden utilizarse para definir el mapeo de los objetos en el conjunto de entrenamiento a las unidades del mapa. Después de la fase de entrenamiento, se puede usar varias funciones de trazado para la visualización; el paquete puede mostrar dónde se asignan los objetos, tiene varias opciones para visualizar los vectores de libro de códigos de las unidades de mapa, y proporciona medios para evaluar el progreso del entrenamiento. Existen funciones de resumen para todo tipo SOM.

IV-B. **Aprendizaje Supervisado**

1. ***Perceptrón Simple:***
   Este modelo tiene gran importancia histórica ya que fue el primer modelo en poseer un mecanismo de entrenamiento que permite determinar automáticamente los pesos sinápticos que clasifican correctamente a un conjunto de patrones a partir de un conjunto de ejemplos.

La arquitectura del perceptrón esta compuesta por dos capas de neuronas, una de entrada y una de salida. La capa de entrada es la que recibe la información proveniente del exterior y la transmite a las neuronas sin realizar ningún tipo de operación sobre la señal de entrada.

En general la información entrante es binaria. La función de activación de las neuronas de un perceptrón es del tipo escalón, dando de esta manera sólo salidas binarias. Cada neurona de salida del perceptrón representa a una clase. Una neurona de salida responde con 1 si el vector de entrada pertenece a la clase a la que representa y responde con 0 en caso contrario.

La operación de un perceptrón con **n** neuronas de entrada y **m** neuronas de salidas puede ser resumida de la siguiente manera:

$$y_i(t) = f(\sum_{j=1}^{n} w_{ij}x_j - \theta_i) \quad \forall i \; 1 \leq i \leq m.$$

El algoritmo de entrenamiento del perceptrón se encuentra dentro de los denominados algoritmos por corrección de errores. Este tipo de algoritmos ajustan los pesos de manera proporcional a la diferencia entre la salida actual proporcionada por la red y la salida objetivo, con el fin de minimizar el error producido por la red.

Ventajas y Desventajas:

Se puede demostrar que este método de entrenamiento converge siempre en un tiempo finito y con independencia de los pesos de partida, siempre que la función a representar sea linealmente separable. El principal problema de este método de entrenamiento es que cuando la función a representar no es linealmente separable el proceso de entrenamiento oscilará y nunca alcanzará la solución. Las funciones no separables linealmente no pueden ser representadas por un perceptrón.

2. ***Perceptron Multicapa***

El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple. El perceptrón multicapa puede ser totalmente o localmente conectado.

- **BackPropagation**
  La propagación hacia atrás de errores o retropropagación es un algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales. El algoritmo emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

  La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

IV-C. Aprendizaje NO Supervisado

Las relaciones que se establecen entre los componentes que describen y definen la información son muy importantes para poder seleccionar posteriormente, de forma inteligente la información que contiene la base de conocimiento. **La clasificación automática de la información, y el modo en que esta es recuperada son los campos donde más se han aplicado las RNA.**

En la redes autoorganizadas el entrenamiento se realiza presentando solo entradas, para ello la red debe de descrubir un patron o características que identifique a cada unas de las entradas. La regla más utilizada por estas redes es el aprendizaje competitivo, de estas se desprenden las siguientes:

- ART (Grossberg 88)
- Neo-cognitrón (Fukushima 80)
- Mapas Autoorganizados (Kohonen 82,89)

IV-C1. ***Mapas Autoorganizativo de KOHONEN****:* Se trata de establecer una correspondencia entre la información de entrada y un espacio de salida de dos dimensiones, o mapa topológico. De esta manera, los datos de entrada con características comunes activarán zonas próximas del mapa.

Este modelo de red representa sus neuronas de salida dispuestas de manera bidimensional. Cuando se ingresa un dato a la red esta reacciona de forma que solo una neurona de la capa de salida resulta activada. A esta neurona se la llama vencedora y determina un punto en el mapa bidimensional. De esta manera se clasifica la información de entrada, ya que la neurona ganadora representa la clase a la que pertenece la entrada, además de que ante entradas similares se activará siempre la misma neurona. Por tanto, la red es sumamente indicada para establecer relaciones, desconocidas previamente, entre un conjunto determinado de datos.

1. **Arquitectura**
   Los mapas autoorganizados consta de dos capas con **N** neuronas de entradas y **M** de salida. Cada una de las **N** neuronas de entrada se conecta a las **M** neuronas de salida.

Entre las neuronas de la capa de salida existen conexiones laterales de inhibicion implícita. Cada una de estas neuronas va a tener cierta influencia sobre su vecina. El valor que se asigne a los pesos de las conexiones entre las capas de entradas y salida durante el proceso de aprendizaje de la red va a deender precisamente de esta interación lateral.

La influencia que una neruona ejerce sobre las demás es función de la distancia entre ellas, siendo muy pequeña cuando están muy alejadas.

2. ***Funcionamiento:***
El algoritmos pretende encontrar el dato aprendido más parecido al de entrada para, en consecuencia, averiguar qué neurona se activará y sobre todo, en qué zona del espacio bidimensional de salida se encuentra.

Lo que hace la red en definitiva es realizar una tarea de clasificación, ya que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información de entrada. Además, como ante otra entrada parecida se activa la misma neurona de salida, u otra cercana, debido a la semejanza entre las clases, se garantiza que las neuronas topológicamente próximas sean sensibles a entradas físicamente similares. Por tanto, esta red es especialmente útil para establecer relaciones entre conjuntos de datos.

3. ***Aprendizaje:***
Es de tipo OFF LINE, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. También utiliza un aprendizaje no supervisado de tipo competitivo. Sólo una neurona de la capa de salida se activa ante la entrada, ajustándose los pesos de las conexiones en función de la neurona que ha resultado vencedora.

Durante la etapa de entrenamiento, se presenta a la red un conjunto de informaciones de entrada para que ésta establezca, en función de la semejanza entre los datos, las diferentes categorías (una por neurona de salida) que servirán durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones entre cada neurona de la capa de salida con las de entrada se corresponderán con los valores de los componentes del vector de aprendizaje que consigue activar la neurona correspondiente.

El aprendizaje no concluye después de presentarle una vez todos los patrones de entrada, sino que habrá que repetir el proceso varias veces para refinar el mapa topológico de salida, de tal forma que cuantas más veces se presenten los datos, tanto más se reducirán las zonas de neuronas que se deben activar ante entradas parecidas, consiguiendo que la red pueda realizar una clasificación más selectiva.

4. ***Algoritmo:***
Sea N el número de neuronas de entrada y M el número de neuronas de salida.
Objetivo: establecer los valores de los pesos de las conexiones.
   a) Inicializar los pesos, w_{ji}, con valores aleatorios pequeños, fijando la zona inicial de vecindad entre las neuronas de salida.

   b) Presentar a la red una **información de entrada** en forma de vector $E_k = (e_1, ..., e_N)$, cuyas componentes, e_i, sean valores contínuos.

   c) Determinar la **neurona vencedora** de la capa de salida: será aquella cuyo vector de pesos, W_j, sea el más parecido a la información de entrada E_k. Recordemos que las componentes de W_j son los valores de los pesos de las conexiones entre esa neurona, j, y cada una de las neuronas de entrada. Para ello se calcula las distancias entre los vectores E_k y W_j para cada neurona de salida. La expresión matemática sería la siguiente:
   $$d_j = \sum_{i=1}^{N}(e_i - w_{ji})^2, 1 \leq j \leq M$$
   d) Una vez localizada la neurona vencedora, j^*, se **actualizan los pesos de sus conexiones** de entrada y también los de las neuronas vecinas (las que pertenecen a su zona de vecindad, Zonaj). Lo que se consigue es asociar la información de entrada con una cierta zona de la capa de salida.
   $$w_{ji}(t+1) = w_{ji}(t) + (t)[e_i^{(k)} - w_{j*i}(t)]$$
   para j perteneciente a Zonaj*(t)
   El término (t) es un parámetro de gananacia o coeficiente de aprendizaje, con un valor entre 0 y 1 que decrece con el número de iteraciones del proceso de entrenamiento. Puede utilizarse la expresión $(t) = 1/t$

   e) El proceso se debe repetir, volviendo a presentar todo el juego de patrones de aprendizaje $E_1, E_2, ...,$ un mínimo de 500 veces (t = 500).

V. *Objetivos*

- Describir los principios, elementos, y funcionamiento de las redes neuronales.
- Conocer la formas de aprendizaje y cómo estas implementan formas distintas para lograr ello.
- Comparar una neurona biológica de una red neuronal artificial.
- Analizar mediante un enfoque distinto el funcionamiento del sistema nervioso mediante la simulación de las redes neuronales.
- Analizar los paquetes lenguaje R para el entrenamiento de las redes neuronales.
- Explicar los diferentes tipos de aprendizaje, tanto supervisado como no supervisado, utilizando los paquetes de

R: **kohonen y neuralnet**.

- Explicar a fondo las siguientes tipos de redes: Perceptron simple, Percetron multicapa y Mapas Autoorganizados(KOHONEN).

VI. *Conclusión*

- El perceptrón simple es útil para cuando se requiere clasificar dos conjuntos linealmente independientes, cuando se requiere de un caso contrario es mejor usar el perceptron multicapa, que en este caso el más comun es el backpropagation.
- El Perceptrón Multicapa no extrapola bien, es decir, si la red se entrena mal o de manera insuficiente, las salidas pueden ser imprecisas.La existencia de mínimos locales en la función de error dificulta considerablemente el entrenamiento, pues una vez alcanzado un mínimo el entrenamiento se detiene aunque no se haya alcanzado la tasa de convergencia fijada.
- Utilizar **lenguaje R** se tiene el beneficio de poder utilizar grandes bases de datos y además de manipularlas.
- Utilizar **paquetes de R** se puede lograr un buen entendimiento de las redes neuronales gracias a sus funciones gráficas que esta posee.

REFERENCIAS

[1] Raquel Flórez López, José Miguel Fernández Fernández. Las Redes Neuronales Artificiales, 2008.

[2] Adam Golda. Principles of Training Multi-Layer Neural Network using Backpropagation, 2005. http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html.

[3] Sr. Luis Federico Bertona. Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos, 2005. http://laboratorios.fi.uba.ar/lsi/bertona-tesisingenieriainformatica.pdf.

[4] Félix de Moya Anegón, Víctor Herrero Solana, Vicente Guerrero Bote. Aplicación de Redes Neuronales Artificiales a la recuperación de información, 1998. https://www.researchgate.net/profile/Felix_Moya-Anegon2/publication/39105442_La_aplicacion_de_Redes_Neuronales_Artificiales_RNA_a_la_recuperacion_de_la_informacion/links/0c96051fded9f08024000000.pdf.

[5] Francisco Gutierrez Martín. Redes Neuronales Artificiales. http://thales.cica.es/rd//Recursos/rd98/TecInfo/07/tecinfo-07.html.

[6] Natividad Noverges, Vicente Sacristán, Pepa Ortí, Lourdes Margaix. Algunas Aplicaciones de Redes Neuronales Artificiales en Documentación, 2000 - 2001. http://personales.upv.es/ccarrasc/doc/2000-2001/Redes%20Neuronales%20Excalibur%20-%20Nativ.%20Noverges/redes.htm#6..

[7] Vaibhav V. Shah, Smitkumar J. Mirani, Yashvardhan V. Nanavati, Vishal Narayanan, Sheetal I. Pereira. Stock Market Prediction using Neural Networks, 2016. http://www.ijsce.org/attachments/File/v6i1/A2816036116.pdf.

[8] Ron Wehrens, Lutgarde M. C. Buydens. Self- and Super-organizing Maps in R: The kohonen Package, 2007. https://core.ac.uk/download/files/153/6303136.pdf.

[9] Frauke Günther, Stefan Fritsch. Neuralnet: Training of Neural Networks, 2010. https://core.ac.uk/download/files/153/6303136.pdf.