Universidad de las Ciencias Informáticas

Facultad 10

Categorización de texto usando Redes Neuronales Artificiales.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autores:

Daimerys Ceballo Gastell

Yanet del Carmen de Diego Ceruto

Tutor:

Ing. Alain Guerrero Enamorado

Co-tutor:

Ing. Mileysis Placencia Crespo

Ciudad de la Habana

Julio de 2008

Declaración de Autoría:

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Daimerys Ceballo Gastell **Yanet del Carmen de Diego Ceruto**

_____ _____

Firma del Autor Firma del Autor

Alain Guerrero Enamorado

_____

Firma del Tutor

Opinión del Tutor del Trabajo de Diploma:

Título: Categorización de texto usando Redes Neuronales Artificiales.

Autores: Daimerys Ceballo Gastell y Yanet del Carmen de Diego Ceruto.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de:

Dedicatoria:

***A mis padres y a mi hermano**, por ser las mejores personas del mundo.*

***A mi novio,** por brindarme todo su amor y su apoyo.*

***A mi familia y amigos,** por su inmenso cariño.*

Daimerys Ceballo Gastell

Dedicatoria:

***A mis padres**: por ser tan especiales, por todo el amor y el apoyo que me han brindado a pesar de la distancia, por el ánimo en los momentos difíciles, por ser mi guía y por confiar siempre en mí.*

***A mi adorable abuelita Ramona**: por ser la persona que alumbra mis pasos, por todo su esfuerzo y su ayuda constante, por quererme tanto y apoyarme siempre.*

***A mi querido hermano Pepe**: por ser mi orgullo y mi paradigma, por todo su apoyo y su esfuerzo, por el amor tan grande que existe entre nosotros, por ser mi guía y mi inspiración.*

***A mi tía Carmen**: por su cariño y ayuda, y por todas las cosas buenas que he aprendido de ti.*

***A mi "viejito" Arsel**: por ser una de las más maravillosas personas que he tenido la oportunidad de conocer, por educarme y enseñarme tanto durante estos cinco años de carrera, por su apoyo y sus consejos, por ser mi amigo y mi otro padre.*

***A Marlen**: por su preocupación y confianza, por su amor y ayuda incondicional, por ser como una madre conmigo.*

***A Yeni**: por todo su apoyo durante estos cinco años, por su confianza y por todo su cariño.*

Yanet de Diego Ceruto.

Agradecimientos:

A mi Mamá y a mi Papá, Magalys y Felix Antonio*: por ser tan especiales, por dedicar gran parte de sus vidas para formarme, por todo su amor y cariño, por estar siempre conmigo, apoyándome, aconsejándome, guiándome, complaciéndome en todos mis caprichos, a ellos les estoy eternamente agradecida.*

A mi Hermano, Daymer: *por enseñarme tanto, por ser fuente de inspiración para alcanzar mis metas, por estar siempre preocupándose por mí, por su cariño y comprensión.*

A Yanet, mi cuñada: *por su ayuda y preocupación.*

A mi novio, Alain*: por cuidarme, ayudarme, complacerme y por regalarme cada día su amor y bondad. Por su inmensa ayuda en la realización de este trabajo y su enorme esfuerzo para que quedara con la calidad requerida.*

A mi compañera de tesis y amiga, Yanet*: por soportarme todos estos años de universidad, por aconsejarme y ayudarme a ser cada día mejor, por su cariño y comprensión, por su disposición y ayuda.*

A mi abuelita Juanita y a mi tía Adelina: *porque siempre están pendientes de todos mis pasos y me adoran como yo a ellas.*

A mi vecinito y amigo Jaisniel: *que siempre esta dispuesto a ofrecerme su ayuda en cualquier circunstancia.*

A mis tíos, Ida, Jeovany, Manuel, Nery, Felix, Blade y Justo así como a sus hijos y esposas*: por toda su preocupación y por el apoyo brindado.*

A mis amiguitas del preuniversitario Midalys, Yanaivys, Nideisys y Yanara*: por poder contar con su cariño, confianza y lealtad.*

A mis amigos del grupo tres y en especial a Lisbet, Arianna, Laira y el Butty: *por hacer que cada pedazo de mi tiempo fuera ameno y por sus buenos consejos.*

A todas aquellas personas que hicieron posible la realización de este trabajo.

Muchísimas Gracias*.*

Daimerys Ceballo Gastell.

Agradecimientos:

A Alain: *mi tutor, por su ayuda inmensa y sus enseñanzas, por todo su esfuerzo para que todo saliera bien, gracias es poco para ti.*

A mi familia: *por todo el esfuerzo y la ayuda que me han brindado, por todo el amor y la preocupación a pesar de la distancia.*

A Arsel, Marlen y Yeni: *mi otra familia, por acogerme en su casa como una hija más y darme toda su confianza y cariño, muchas gracias.*

A Dayme*: por ser una de las personas que más admiro y que con su sinceridad y su cariño me ha enseñado a ser mejor, por ser mi hermana y estar siempre cuando te he necesitado.*

A mis amigos Walter, Russo y Chinese: *por romper las barreras de la distancia, por su preocupación y todo el apoyo que me han brindado, los quiero mucho.*

A Arianna, Laira y Lisbet: *por ser como mi familia, conocerlas ha significado mucho para mí, gracias por todos los momentos que hemos pasado juntas y que serán inolvidables.*

A la familia de Day*: por toda la ayuda y el amor que me han brindado.*

A Yiselita*: mi amiga de la secundaria, por todo su cariño y preocupación a pesar de la distancia.*

A Jeizen: *por ser una de esas personas que llega para quedarse, gracias por todas las cosas que me enseñaste y los momentos geniales que siempre permanecerán en mi memoria.*

A los muchachos de mi grupo de la facultad 2*: por todos los momentos que hemos compartido y que no se olvidarán.*

A Darien: *por sus enseñanzas y su preocupación.*

A Kike, Botico y Yasmani*: por su amistad sincera.*

A Yudel*: por toda su ayuda y preocupación.*

A Surima*: que aunque llegó tarde, es de esas personas que tampoco se van.*

A Dionny*: por hacerme volar en estos tiempos en que casi se pierden las alas.*

A Angelito y Robinson: *a quienes tantas veces molesté, gracias por su ayuda durante estos cinco años.*

A Jose, Kenny, Leo, Yuri y David*: los muchachos de Ucistore con quienes tanto he disfrutado de uno de mis mayores placeres, la música.*

A todas la personas que he tenido el placer de conocer en la universidad y que de alguna manera han influido en el desarrollo de este trabajo y se han convertido en muy buenos amigos, ***mis más sinceros agradecimientos****.*

Yanet de Diego Ceruto.

Resumen:

Durante mucho tiempo, la dinámica de Internet, ha dificultado en el proyecto FILPACON (Filtrado de Paquetes por Contenido) contar con una base de datos de contenidos categorizados lo suficientemente actualizada. En este sentido, se concibió como objetivo de esta tesis obtener un modelo usando Redes Neuronales Artificiales (RNA) que permitiera la categorización de texto, de esta manera, se lograría automatizar el proceso de actualización de la base de datos. Una investigación previa sobre las RNA en general y aplicadas a la categorización de texto más tarde, creó las bases teóricas que permitieron sustentar los fundamentos y la propuesta final del presente trabajo. A tal efecto, se seleccionó un algoritmo de entrenamiento eficiente para la categorización de texto, se creó manualmente una colección de entrenamiento que incluyó cinco categorías, elegidas de acuerdo a la necesidad del proyecto y se concluyó que es viable la aplicación del modelo propuesto ya que se obtuvo una arquitectura de red y unos parámetros de ajuste que arrojaron óptimos resultados en la tarea de categorización de texto.

Palabras Clave: Redes Neuronales Artificiales, Aprendizaje por Cuantificación Vectorial (LVQ), Categorización de Texto, FILPACON.

Abstract:

During a long time, the dynamics of Internet has slowed down the process of having a categorized contents database sufficiently updated in the project FILPACON (Filtering of Packages by Content). Arising from this fact, it was conceived as objective of this thesis work to obtain a model using Artificial Neuronal Nets (ANN) that allowed the text categorization, this way, it would be possible to automate the process of updating the database. A previous investigation on the ANN in general and later applied to the text categorization, created the theoretical grounds that allowed sustaining the foundations and the final proposal of the present work. To such an effect, an algorithm of efficient training was selected for the text categorization, it was manually created a training collection which included five categories, selected according to the project needs and it was concluded that the application of the model proposed is feasible since it was obtained a net architecture and some adjustment parameters which granted optimal results while carrying out the task of text categorization.

Keywords: Artificial Neuronal Nets, Learning Vector Quantization (LVQ), Text Categorization, FILPACON.

Índice de Contenido:

Índice de Figuras:

Índice de Tablas:

Introducción

La red de redes -Internet- alberga una inmensa variedad de contenidos y ofrece muchísimas posibilidades para informarse, educarse, comerciar y entretenerse. Su creciente importancia la ha convertido en una de las principales palancas del mundo moderno. Pero pese a todo esto, el uso de Internet también conlleva riesgos, ya que de igual forma existen en la red innumerables contenidos de cuestionable valor moral o educativo y muchos otros que violan leyes de disímiles países. En su filosofía Internet ha sido diseñada como una red distribuida, es decir, que abarca muchas redes voluntariamente interconectadas y como tal, no tiene ningún cuerpo que la gobierne; esto dificulta que, por ejemplo, se pueda definir de manera global qué es ilícito, qué es nocivo y qué es adecuado para los usuarios, pues esto dependería de factores culturales, políticos, religiosos y otros muchos que varían de una nación a otra, y en ocasiones, de una organización a otra.

El Sistema de Filtrado de Contenidos para Internet (FILPACON), pretende ser una solución factible a la hora de proteger a determinados sectores de usuarios mientras navegan por Internet, de manera que el ciberespacio pueda ser zonificado en dependencia de los contenidos que albergue y permitiendo que los administradores en conjunto con los usuarios de las organizaciones puedan definir qué debe pertenecer a un determinado tipo de contenido. Sin embargo el sistema debe resolver un problema adicional; la dinámica de Internet es tal que según estimaciones[1] aparece una nueva página cada 0.75 segundos, lo cual hace difícil la tarea de contar con una base de datos de contenidos categorizados lo suficientemente actualizada si se hiciera de manera manual. Dada esta situación y teniendo en cuenta que en este punto la Inteligencia Artificial (IA) brinda una amplia gama de técnicas para categorizar contenidos correctamente, se hace necesario realizar pruebas de categorización de texto utilizando Redes Neuronales Artificiales (RNA), para proponer un modelo que pudiera resultar eficiente en lo que a la categorización de texto se refiere. Para ello se ha determinado como **problema científico**:

¿Cómo contribuir a la categorización de texto mediante un modelo basado en Redes Neuronales Artificiales para el proyecto FILPACON?

[1] http://news.netcraft.com/

Partiendo del problema enunciado anteriormente se define como **objeto de estudio:** las Redes Neuronales Artificiales, donde el **campo de acción** estaría delimitado por el estudio de las RNA aplicadas a la categorización de texto para FILPACON.

De manera general el **objetivo es:**

Proponer un modelo usando Redes Neuronales Artificiales que permita la categorización de texto para el Sistema de Filtrado de Contenidos (FILPACON).

Para complementar este objetivo general se puntualizan los siguientes **objetivos específicos:**

- Seleccionar un algoritmo de entrenamiento eficiente para la categorización de texto.
- Seleccionar un modelo de representación textual para la categorización de texto usando Redes Neuronales Artificiales.
- Crear una colección de entrenamiento que incluya al menos tres categorías.
- Determinar el mejor modelo de Red Neuronal Artificial.

Para lograr los objetivos propuestos se formulan las siguientes **preguntas científicas:**

- ¿Cuáles algoritmos se utilizan para entrenar Redes Neuronales en la categorización de texto?
- ¿Cuáles modelos de representación textual pueden utilizarse para la categorización de texto usando Redes Neuronales Artificiales?
- ¿Cómo escoger una colección de documentos que sirva para entrenar y validar el modelo de red que se proponga?
- ¿Qué software se utiliza para simular Redes Neuronales Artificiales?

Para dar cumplimiento de manera exitosa a los objetivos planteados se deben desarrollar las siguientes **tareas**:

- Estudiar el funcionamiento de las Redes Neuronales Artificiales para permitir la selección de un algoritmo de entrenamiento eficiente para la categorización de texto.
- Realizar una búsqueda del modelo de representación textual más factible para la categorización de texto usando Redes Neuronales.

- Seleccionar tres o más de las categorías representativas del proyecto FILPACON y escoger en Internet varios documentos por cada una de estas para crear la colección de entrenamiento.
- Entrenar la red neuronal con el 75% de la colección y probarla con el 25% restante.
- Realizar experimentos con diferentes topologías para proponer un modelo óptimo para la categorización de texto.

Una vez concluido el trabajo se espera se pueda contar con los siguientes resultados:

- Colección de entrenamiento, para cada categoría que se seleccione, pre-categorizada manualmente.
- Modelo de red neuronal eficiente para categorizar texto.
- Base teórica para la implementación de una aplicación de categorización de texto usando el modelo de red neuronal propuesto.

Para facilitar el alcance de estos resultados y una mayor organización se emplearán los siguientes **métodos científicos de investigación:**

Métodos teóricos.

- Analítico-Sintético: permite hacer un análisis del funcionamiento de las Redes Neuronales y su aplicación a la categorización de texto, que posibilita descubrir sus características generales y las relaciones esenciales entre ellas, obteniéndose elementos necesarios para aplicar al desarrollo de la investigación.
- Inducción-Deducción: para analizar las características del comportamiento de los modelos neuronales en la categorización de texto y así poder deducir conclusiones sobre casos particulares que pueden ser verificados en la práctica.
- Histórico-Lógico: para estudiar la evolución y desarrollo de los modelos neuronales en la categorización de texto y comprender lógicamente como serían sus tendencias actuales.

Métodos empíricos.

- Observación: mediante la observación se pueden analizar los resultados e investigar como se comporta el modelo neuronal.

- Medición: para obtener información numérica acerca del comportamiento de la red neuronal y comparar sus valores.
- Experimentación: por su importancia decisiva en la demostración del comportamiento del modelo de red neuronal.

El presente documento está compuesto por tres capítulos, estructurados de la siguiente manera:

Capítulo 1: Redes Neuronales Artificiales.

En este capítulo se realiza un estudio de las Redes Neuronales Artificiales donde se tratan importantes temas relacionados con funcionamiento, arquitectura, características y clasificaciones según diferentes criterios, etc. Finalmente se comentan algunas de las aplicaciones de las Redes Neuronales Artificiales a la categorización de texto.

Capítulo 2: Representación de documentos y modelo neuronal de Kohonen.

En este capítulo se describe como se procesan los documentos antes de dárselos como entrada al clasificador, explicando los mecanismos usados para realizar esta tarea como son las listas de parada y el algoritmo de Porter. Además del modelo de representación textual seleccionado, también se presenta el modelo neuronal de Kohonen en su variante supervisada y por último se especifican las métricas de evaluación que se emplearán.

Capítulo 3: Algoritmo LVQ aplicado a la categorización de texto.

Este capítulo describe y muestra los resultados de los experimentos realizados con el algoritmo propuesto (LVQ), aplicado a la categorización de texto; y se determinan los parámetros para los que la red obtiene el mayor porciento de aciertos.

Capítulo 1 Redes Neuronales Artificiales.

En este capítulo se presenta un estudio de las Redes Neuronales Artificiales que sirve de base para la comprensión de los posteriores capítulos ya que el modelo que se propone para categorizar texto es un modelo neuronal.

Se hace un acercamiento a los principales conceptos de Redes Neuronales Artificiales. Se abordan aspectos tales como funcionamiento, arquitectura, características y clasificaciones de las redes según diferentes criterios. También se describen las ventajas que tienen, así como algunas de las principales áreas de aplicaciones. El último apartado se dedica a las Redes Neuronales Artificiales en la categorización de texto, donde, de manera resumida, se hace referencia a algunos trabajos dedicados a este tema a nivel internacional. Se hace énfasis en los algoritmos utilizados por los diferentes autores, para finalmente presentar el algoritmo que se ha seleccionado para realizar la categorización de texto.

1.1. Introducción.

Es evidente que hoy en día las computadoras son de gran ayuda en tareas que resultan realmente engorrosas para el hombre, ya que pueden resolver de forma automática y rápida determinadas operaciones que muchas veces son tediosas cuando se realizan manualmente. La evolución del hardware ha hecho que la potencia de cálculo haya crecido de tal forma que los ordenadores se han vuelto indispensables en muchas áreas de actividad del ser humano, sin embargo la computación algorítmica no es suficiente cuando se han de enfrentar ciertas tareas. Por ejemplo, algo tan sencillo para el ser humano como reconocer una cara de otra persona es el tipo de problema que no es tan fácil ser resuelto por la vía algorítmica (BURGOS, 2003).

Debido a este tipo de situaciones desde finales de los 50 se ha venido investigando en un conjunto de técnicas que utilizan un enfoque diferente para resolver los problemas. Este conjunto de técnicas y herramientas se bautizó con el nombre de Inteligencia Artificial (IA), porque lo que se pretendía era que los ordenadores presentaran un comportamiento inteligente, entendiendo por esto que supieran hacer frente a ciertos problemas de una manera similar a como lo hacen los seres humanos.

La IA al tratar de construir máquinas que se comporten aparentemente como seres humanos ha dado lugar al surgimiento de dos enfoques distintos (BURGOS, 2003). Por un lado, se desarrolló lo que se conoce como el enfoque simbólico o top-down, conocido como la IA clásica. Este enfoque asienta sus bases en la manipulación de símbolos en vez del mero cálculo numérico, tradicional de la computación algorítmica. La realidad se plasma por medio de una serie de reglas. Herramientas como la lógica de predicados, permiten manipular los símbolos y las reglas para obtener nuevas reglas.

El otro enfoque es el enfoque subsimbólico, llamado a veces conexionista. Sus esfuerzos se orientan a la simulación de los elementos de más bajo nivel dentro de los procesos inteligentes con la esperanza de que estos al combinarse permitan que espontáneamente surja el comportamiento inteligente.

En este último enfoque es donde se encuadran las Redes Neuronales Artificiales. Estas no 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 son un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto del que se dispone para un sistema que es capaz de adquirir conocimiento a través de la experiencia. Una red neuronal es un sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona (KAFATI, 2008).

1.1.1 Funcionamiento de una neurona biológica.

La neurona es la unidad básica de procesamiento del cerebro el cual consta de un gran número (aproximadamente 100 000 millones) altamente interconectadas (entre 1000 y 10 000 conexiones por elemento)(FERRÚS, 2007). Cada célula nerviosa o neurona consta de una porción central o cuerpo celular, que contiene el núcleo y una o más estructuras denominadas axones y dendritas.

Estas últimas son unas extensiones cortas y ramificadas que forman una especie de árbol alrededor del cuerpo neuronal, constituyen la principal superficie sobre la cual la neurona recibe los estímulos. En contraste, el axón suele ser una prolongación única y alargada y a diferencia de las dendritas las ramificaciones de este se encuentran justo al final de la fibra, lugar donde ocurre la comunicación con otras neuronas. La zona de contacto entre un axón de una célula y una dendrita de otra célula es llamada sinapsis. (Figura 1.1)

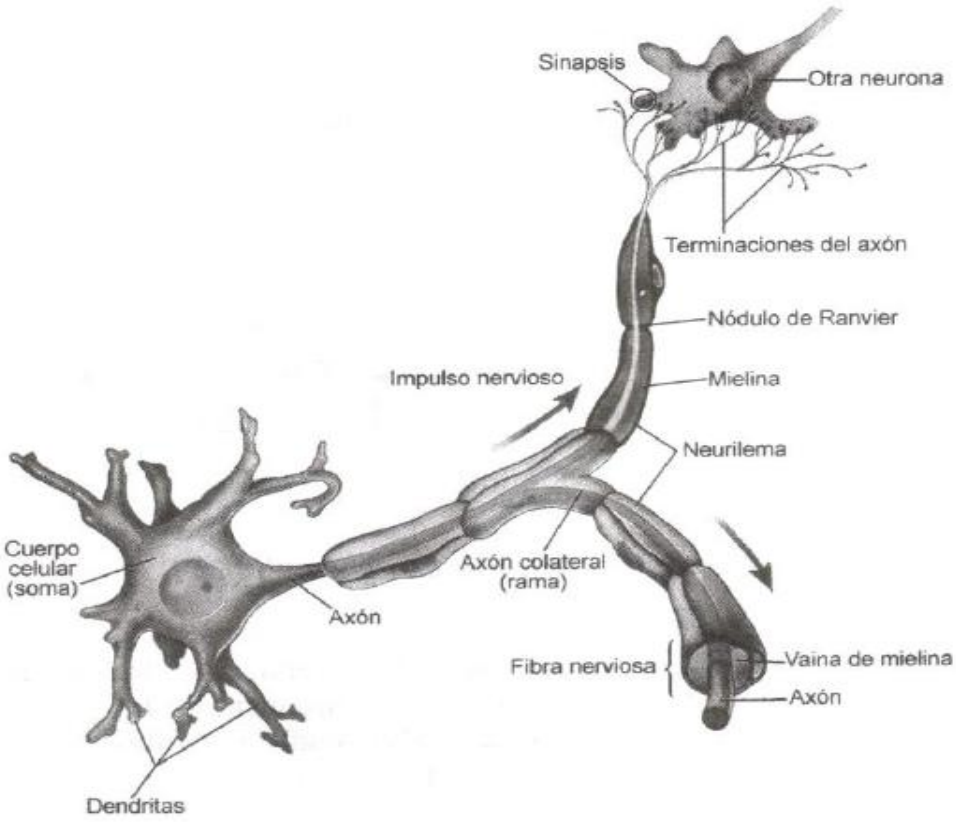

Figura 1. 1: Neurona Biológica.

La membrana exterior de las neuronas tiene propiedades especiales, a lo largo del axón la membrana está especializada en propagar el impulso nervioso, esto se logra a través de impulsos eléctricos, denominados potenciales de acción, que alcanzan una amplitud máxima de unos 100 mv y duran 1 ms. Estos son el resultado del desplazamiento a través de la membrana celular de iones de sodio dotados de carga positiva, que pasan desde el fluido extracelular hasta el citoplasma intracelular; la concentración extracelular de sodio es mayor que la concentración intracelular, la carga positiva inyectada en el axón durante el potencial de acción queda disipada

uno o dos milímetros más adelante, para que la señal recorra varios centímetros es preciso regenerar frecuentemente el potencial de acción a lo largo del camino; la necesidad de reforzar repetidamente esta corriente eléctrica limita a unos 100 metros por segundo la velocidad máxima de viaje de los impulsos, tal velocidad es inferior a la millonésima de la velocidad de una señal eléctrica por un hilo de cobre.

Estas señales de baja frecuencia (potenciales de acción) no pueden saltar de una célula a otra, la comunicación entre neuronas viene siempre mediada por neurotransmisores químicos que son liberados en el espacio sináptico.

Cuando un potencial de acción llega al terminal de un axón son liberados los neurotransmisores alojados en diminutas vesículas, que después son vertidos en una hendidura de unos 20 nanómetros de anchura que separa la membrana presináptica de la postsináptica; durante el apogeo del potencial de acción, penetran iones de calcio en el terminal nervioso, su movimiento constituye la señal determinante de la exocitosis sincronizada, esto es la liberación coordinada de moléculas neurotransmisoras. En cuanto son liberados, los neurotransmisores se enlazan con receptores postsinápticos, instando el cambio de la permeabilidad de la membrana. Cuando el desplazamiento de carga hace que la membrana se aproxime al umbral de generación de potenciales de acción, se produce un efecto excitador y cuando la membrana resulta estabilizada en la vecindad el valor de reposo produce un efecto inhibidor.

Cada sinapsis produce sólo un pequeño efecto, para determinar la intensidad (frecuencia de los potenciales de acción) de la respuesta cada neurona ha de integrar continuamente hasta unas 1000 señales sinápticas, que se suman en el cuerpo de la célula. Otro aspecto importante de la membrana es que media en el reconocimiento de las otras células durante el desarrollo embrionario, de modo que cada célula encuentra su lugar apropiado en la red. Algunas de las estructuras neuronales son determinadas en el nacimiento, otra parte es desarrollada a través del aprendizaje, proceso en que nuevas conexiones neuronales son realizadas y otras se pierden por completo.

Aunque los detalles de operación de las Redes Neuronales Artificiales difieren bastante de los cerebros humanos, estas son similares en tres aspectos:

1. Una Red Neuronal Artificial está formada por un gran número de simples elementos de procesamiento (neuronas).
2. Cada neurona se conecta a otras muchas neuronas.
3. La funcionalidad de la red neuronal viene determinada por la modificación de los pesos de conexión durante la fase de aprendizaje.

1.2. Redes Neuronales Artificiales.

1.2.1 La neurona artificial.

Los estudios sobre el cerebro han llevado a la conclusión de que las neuronas son elementos analógicos. A partir de las entradas sinápticas, el cuerpo de la neurona hace una suma de estas y genera una salida, la cual es casi siempre un valor continuo, llamado potencial de acción.

El modelo de una neurona artificial es una imitación del proceso de una neurona biológica, puede también parecerse a un sumador hecho con un amplificador operacional (ZULUAGA, 2000) tal como se ve en la figura 1.2.

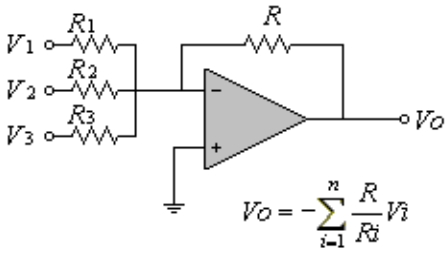

Figura 1. 2: Neurona Artificial.

Una Red Neuronal Artificial es un procesador paralelo distribuido, inspirado en las redes neuronales biológicas, que puede almacenar conocimiento experimental y hacerlo disponible para su uso (PÉREZ, 2005). De manera similar a una red neuronal biológica el conocimiento es adquirido por medio de un proceso de aprendizaje y es almacenado por medio de los pesos sinápticos que representan las conexiones entre neuronas. Una red neuronal se entrena para una determinada tarea, por medio de un algoritmo de aprendizaje, cuya función es modificar los pesos sinápticos para satisfacer un criterio de desempeño especificado.

La neurona es la unidad de proceso de información fundamental en una red neuronal. En el diagrama en bloques de la figura 1.3, se muestra el modelo de una neurona; esta forma la base para el diseño de una Red Neuronal Artificial, se observa una neurona con R entradas; las entradas individuales $P_1, P_2, ..., P_R$ son multiplicadas por los pesos correspondientes $W_{1,1}, W_{1,2}, ..., W_{1,R}$ pertenecientes a la matriz de pesos W .

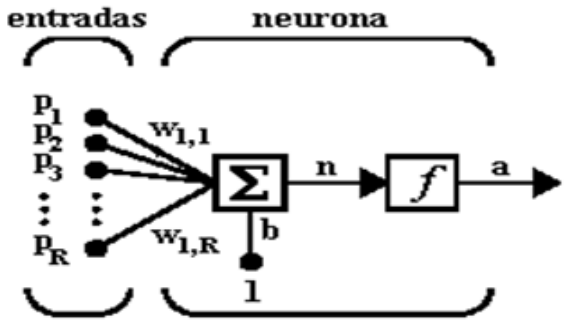

Figura 1. 3: Neurona con múltiples entradas.

La neurona tiene una ganancia b, la cual llega al mismo sumador al que llegan las entradas multiplicadas por los pesos, para formar la salida total n ,

$$n = w_{1,1} p_1 + w_{1,2} p_2 + ... + w_{1,E} p_R + b \qquad (1.1)$$

Esta expresión puede ser escrita en forma matricial

$$n = W_p + b \qquad (1.2)$$

$$a = f(n) \qquad (1.3)$$

1.2.2 Definición de Red Neuronal Artificial.

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. Algunas de estas definiciones se verán a continuación.

Las Redes Neuronales Artificiales son modelos que intentan emular el funcionamiento cerebral o la organización neuronal. Básicamente una Red Neuronal Artificial [RNA] consiste en un conjunto de unidades computacionales simples o elementos de procesamientos o “nodos”, (símil neurona), y un conjunto de conexiones que unen dichas unidades (representado en forma genérica en la Figura. 1.4) que llamaremos "pesos" (BARRO, 1995; HAYKIN, 1999).

Darpa (DARPA, 1988), define una red neuronal como un sistema compuesto de muchos elementos simples de procesamiento los cuales operan en paralelo y cuya función es determinada por la estructura de la red, el peso de las conexiones; realizándose el procesamiento en cada uno de los nodos o elementos de cómputo.

Como definición formal de Redes Neuronales Artificiales se puede tomar lo expresado por Hayking (HAYKIN, 1994).

Una Red Neuronal Artificial es un procesador paralelo y distribuido que tiene la facilidad natural para almacenar conocimiento experimental y hacerlo útil para su uso, asemejando al cerebro en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso de aprendizaje.
- La “fuerza” de las conexiones inter-neurona, conocida como pesos sinápticos son utilizados para almacenar el conocimiento.

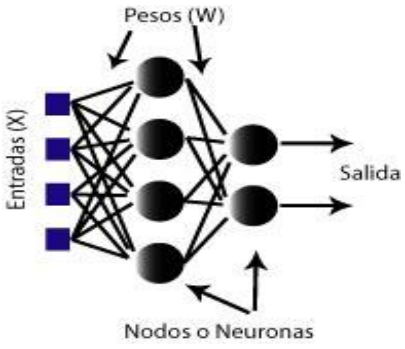

Figura 1. 4: Red Neuronal Artificial.

Otras definición es la propuesta por Hecht Nielsen (NIELSEN, 1990) para el que las Redes Neuronales Artificiales son simulaciones de estructuras cognitivas de procesamiento de información, basadas en modelos de las funciones cerebrales.

Teuvo Kohonen (KOHONEN, 1995) define las Redes Neuronales Artificiales como estructuras interconectadas de unidades de procesamiento simples (generalmente adaptables). Estos dispositivos son masivamente paralelos y de organización jerárquica. Deben interactuar con los objetos del mundo real de la misma forma en que lo hace el sistema nervioso biológico.

Aunque no existe una definición aceptada universalmente puede decirse de manera general que las Redes Neuronales Artificiales son modelos matemáticos multiparamétricos no-lineales, capaces de inducir una correspondencia entre conjuntos de patrones de información (la relación estímulo-respuesta)(VALDIVIA, 2004). El procesamiento de la información se realiza a la manera de los sistemas neuronales biológicos.

Una red neuronal puede verse como un grafo dirigido con las siguientes características(VALDIVIA, 2004):

- Los nodos del grafo se denominan elementos de procesamiento, unidades o neuronas.
- Las uniones entre los nodos se denominan conexiones. Cada conexión funciona en un momento determinado en una única dirección. A cada conexión se le asigna un factor de ponderación denominado peso o intensidad de conexión. Las conexiones pueden ser excitatorias o inhibitorias.
- Cada elemento puede recibir cualquier número de conexiones de entrada procedentes del resto de elementos de procesamiento que forman la red.
- Cada elemento de procesamiento tendrá una única salida que se podrá ramificar para ser aplicada a muchos otros elementos de procesamiento. Así, la salida de una unidad se tomará como entrada a cada uno de los elementos de procesamiento conectados con dicha unidad.
- Cada elemento de procesamiento puede tener una memoria local.
- Cada elemento de procesamiento posee una función de activación que usa la memoria local y las señales de entrada para producir una señal de salida. La salida producida por un elemento de procesamiento depende exclusivamente de los valores actuales de entrada que le llegan de otros elementos de procesamiento a través de las conexiones y los valores almacenados en su memoria local, es decir, el procesamiento de información en cada elemento de procesamiento es completamente local.

1.2.3 Funcionamiento de una Red Neuronal Artificial.

En cuanto al modo interno de trabajo, las redes neuronales son modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez. Estos modelos matemáticos que emplean las redes son multivariantes que utilizan procedimientos iterativos, en general para minimizar funciones de error (ESPINOZA, 2002). Como se ha dicho en el apartado anterior se suelen representar mediante grafos, llamados en este contexto neuronas artificiales. Cada neurona realiza una función matemática. Las neuronas se agrupan en capas, constituyendo una red neuronal. Una determinada red neuronal es confeccionada y entrenada para llevar a cabo una labor específica.

Los modelos neuronales utilizan varios algoritmos de estimación, aprendizaje o entrenamiento para encontrar los valores de los parámetros del modelo, que se denominan pesos sinápticos.

Originalmente la red neuronal no dispone de ningún tipo de conocimiento útil almacenado. Para que la red neuronal ejecute una tarea es preciso entrenarla, en terminología estadística se diría que es necesario estimar los parámetros.

En realidad todo el procedimiento es estadístico: primero se selecciona un conjunto de datos, o patrones de aprendizaje. Después se desarrolla la arquitectura neuronal, número de neuronas, tipo de red; es decir, se selecciona el modelo y el número de variables dependientes e independientes. Se procede a la fase de aprendizaje o estimación del modelo y a continuación se validan los resultados.

Existen dos fases en toda aplicación de las redes neuronales (SKAPURA, 1991):

Fase de entrenamiento: Durante esta fase la red ajusta los pesos de conexión entre las distintas unidades siguiendo un determinado algoritmo de aprendizaje. El objetivo del entrenamiento es encontrar un conjunto de pesos, para los que la aplicación, dado un conjunto de vectores de entrada genere el conjunto de salida deseado. Para que la red aprenda se necesitan vectores de entrenamiento que sean lo suficientemente significativos de lo que se quiere que la red reconozca. Estos vectores serán aplicados a las unidades de entrada y la señal se propagara a través de la

red produciendo una salida. Dependiendo del valor de salida obtenido, los pesos de conexión se modificarán de acuerdo con algún algoritmo de entrenamiento.

Fase de prueba: Una vez que la red se estabiliza, es decir, cuando no hay modificación de los pesos, comienza la explotación de la red. Se aplican unos valores de entrada, se propaga la señal a través de la red y se obtienen los valores de salida.

1.3. Tipos de Redes Neuronales Artificiales.

Existe una gran variedad de Redes Neuronales Artificiales. A continuación se describen aspectos fundamentales de las redes neuronales como los niveles de neuronas, las formas de conexión entre las capas, así como niveles en la arquitectura, dentro de la cual se encuentran distintas clasificaciones atendiendo a diferentes criterios. También se explican las funciones base y de activación y los tipos de aprendizaje y de entrada. La variación de algunos de estos parámetros es lo que contribuye a que hoy en día existan gran variedad de estructuras de Redes Neuronales Artificiales.

1.3.1 Niveles de Neuronas.

La distribución de neuronas dentro de la red se realiza formando niveles o capas de un número determinado de neuronas cada una. A partir de su situación dentro de la red se pueden distinguir tres tipos de capas:

- **De entrada:** las neuronas de esta capa reciben los datos que se proporcionan a la RNA para que los procese.
- **Ocultas:** son capas que sirven para procesar información y comunicar otras capas.
- **De salida:** estas envían la información hacia el exterior, proporciona la respuesta de la red neuronal. Normalmente también realiza parte del procesamiento.

1.3.2 Forma de Conexión de las Capas.

Las neuronas se conectan unas a las otras usando sinapsis. Si se mira detenidamente se puede observar que estas uniones a nivel de capa forman distintas estructuras. Se pueden distinguir varias como:

Unión Todos con Todos:

Consiste en unir cada neurona de una capa con todas las neuronas de la otra capa. Este tipo de conexionado es el más usado en las redes neuronales, se usa en todo tipo de uniones desde el Perceptrón Multicapa a las redes de Hopfield.

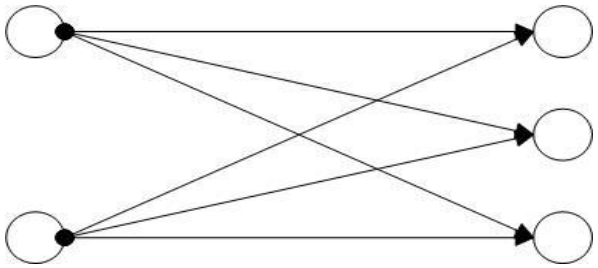

Figura 1. 5: Unión todos con todos.

Unión Lineal:

Consiste en unir cada neurona con otra neurona de la otra capa. Este tipo de unión se usa menos que el anterior y suele usarse para unir la capa de entrada con la capa procesamiento, si la capa de entrada se usa como sensor. También se usa en algunas redes de aprendizaje competitivo

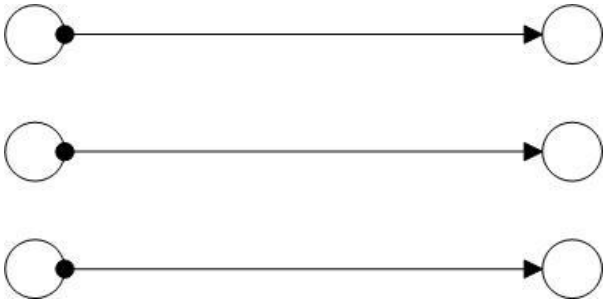

Figura 1. 6: Unión lineal.

Predeterminado:

Este tipo de conexionado aparece en redes que tienen la propiedad de agregar o eliminar neuronas de sus capas y de eliminar también conexiones.

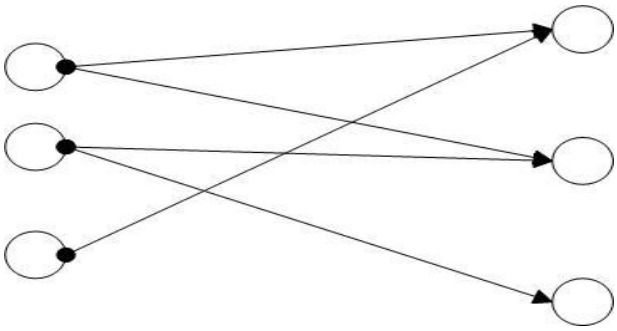

Figura 1. 7: Unión predeterminada.

1.3.3 Niveles en la arquitectura.

La arquitectura de una red consiste en la organización y disposición de las neuronas en la red. Las neuronas se agrupan formando capas, que pueden tener muy distintas características. Además las capas se organizan para formar la estructura de la red. Esto se puede observar en la siguiente figura (1.8).

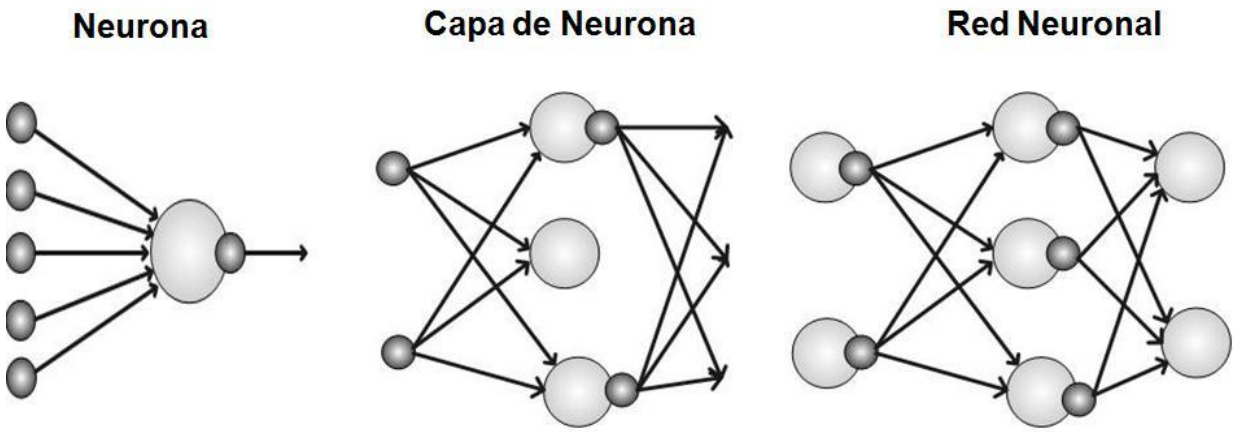

Figura 1. 8: Jerarquía de Redes Neuronales Artificiales.

Los niveles en la arquitectura son tres (GPDS, 2001), a continuación se describen detalladamente.

1.3.3.1 Microestructura

Hace referencia a los elementos más pequeños de las redes neuronales: las neuronas, que pueden tener diferentes formas dependiendo de la aplicación.

1.3.3.2 Mesoestructura

Es el resultado de la combinación de las neuronas. Esta se puede realizar de formas diferentes, siendo aquí donde se habla de capas, y dependiendo del número de éstas y de la conexión entre ellas se tienen diferentes clasificaciones. Estas quedan expuestas en la siguiente tabla (1.1) y se explican a continuación. A la hora de definir la arquitectura de una red neuronal, normalmente se hace referencia a las tres características.

Dependiendo De:	Clasificación:
Número de capas	Monocapa(1capa) Multicapa(más de una capa)
Transmisión de la información	Con conexiones hacia adelante Con conexiones hacia atrás Con conexiones laterales
Número de conexiones	Totalmente Conectada Parcialmente conectada

Tabla 1. 1: Clasificaciones de la Redes Neuronales.

- **Clasificación según el número de capas.**

Monocapa: Las redes monocapa son redes con una sola capa. Para unirse las neuronas crean conexiones laterales para conectar con otras neuronas de su capa. Han sido ampliamente utilizadas en circuitos eléctricos ya que debido a su topología, son adecuadas para ser implementadas mediante hardware, usando matrices de diodos que representan las conexiones de las neuronas.

Multicapa: Las redes multicapa están formadas por varias capas de neuronas, dos o más.

- **Clasificación según se transmite la información.**

Redes feedforward, unidireccional o con conexiones hacia adelante: Este tipo de redes contienen solo conexiones entre capas hacia delante. Esto implica que una capa no puede tener conexiones a una que reciba la señal antes que ella en la dinámica de la computación.

Redes feedback, retroalimentadas o con conexiones hacia atrás: Aparte del orden normal algunas capas están también unidas desde la salida hasta la entrada en el orden inverso en que viajan las señales de información. Este tipo de redes se diferencia de las anteriores en que si pueden existir conexiones de capas hacia atrás y por tanto la información puede regresar a capas anteriores en la dinámica de la red.

Redes feedlateral o con conexiones laterales: son conexiones entre neuronas de la misma capa, este tipo de conexión son muy comunes en las redes mono capa.

- **Clasificación según el número de conexiones.**

Redes neuronales totalmente conectadas. En este caso todas las neuronas de una capa se encuentran conectadas con las de la capa siguiente (**redes no recurrentes**) o con las de la anterior (**redes recurrentes**).

Redes neuronales parcialmente conectadas. En este caso no se da la conexión total entre neuronas de diferentes capas.

1.3.3.3 Macroestructura

Es la combinación de redes, se podría denominar a este nivel “comité de expertos”. Existen problemas donde una combinación de redes da un mejor comportamiento que usar una sola red. Esta combinación puede ser:

- **En paralelo**: todas tienen el mismo peso.
- **En serie**: la salida de una red es la entrada a otra mayor.
- **Jerárquica**: en problemas de clasificación, existen redes más expertas que otras.

Pueden existir otras o variaciones de ellas dependiendo de la aplicación concreta.

1.3.4 Funciones.

Un modelo que facilita el estudio de una neurona, puede visualizarse en la figura 1.9. Se tiene, en primer lugar, los *inputs* o *entradas*, x_n. En segundo lugar, las ponderaciones, β_{ij}, que representan la intensidad de la interacción entre neuronas, es decir, indicadores del conocimiento retenido. En tercer lugar, la *regla de propagación,* $h_i(t) = \sigma(\beta_{ij}, x_j(t))$ y la función de *activación* o *transferencia*, $g_i(h_i(t))$ (que suele ser determinista, monótona creciente y continua). En último lugar, la función de *salida*, $f_i(g_i(h_i(t))) = f_i(g[(\sigma(\beta_{ij}, x_j(t)))])$ (TORRA-P, 2004).

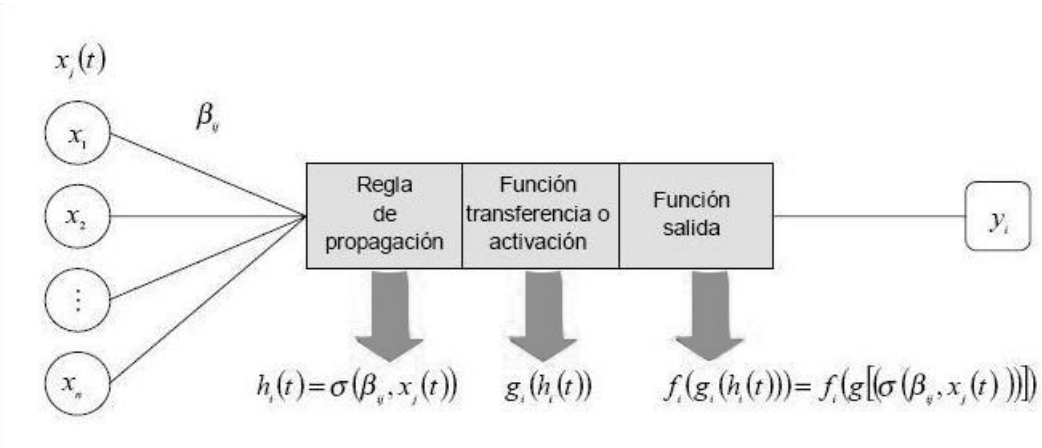

Figura 1. 9: Modelo genérico de *neurona artificial.*

La regla de propagación es un elemento relevante, que se encarga de transformar las diferentes entradas que provienen de la sinapsis en el potencial de la neurona.

Una regla de propagación utilizada es la distancia euclídea. Usada en los mapas de Kohonen y algunas redes competitivas.

En la figura 1.10 se puede observar que la regla de propagación puede poseer diferentes formas. En primer lugar, cuadrática, en segundo lugar, modelos basados en el cálculo de distancias entre vectores, como por ejemplo, la *distancia euclídea*, en tercer lugar, de carácter no lineal como la expresión *polinómica* y por último, la *lineal*.

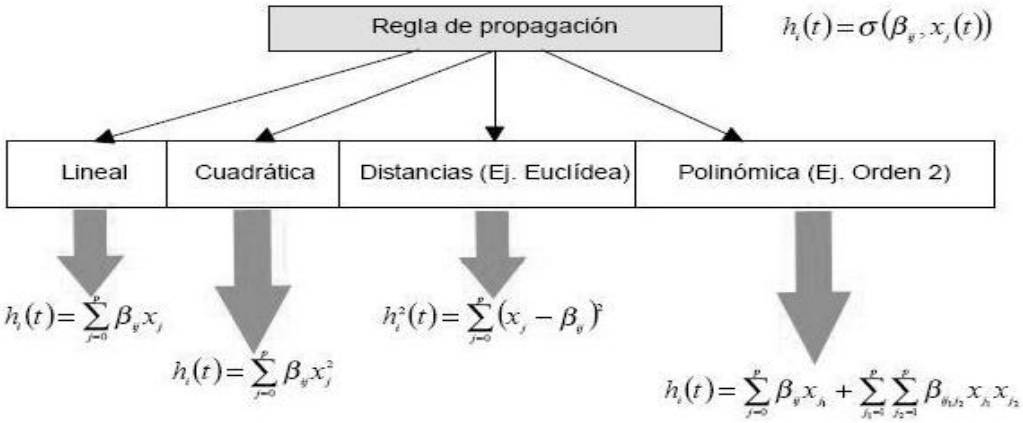

Figura 1. 10: Modelos de *reglas de propagación.*

El valor que surge de la regla de propagación elegida debe ser con posterioridad filtrado mediante la función de transferencia o activación, $g(.)$. Esta función se encarga de calcular el nivel de activación de la neurona en función de la entrada total, también denota la salida de la neurona. En la tabla 1.2 se muestran las principales funciones de transferencia empleadas en el entrenamiento de redes neuronales.

Nombre	Relación Entrada /Salida	Icono	Función
Limitador Fuerte	$a = 0\ n < 0$ $a = 1\ n \geq 0$		*hardlim*
Limitador Fuerte Simétrico	$a = -1\ n < 0$ $a = +1\ n \geq 0$		*hardlims*
Lineal Positiva	$a = 0\ n < 0$ $a = n\ 0 \leq n$		*poslin*
Lineal	$a = n$		*purelin*

Lineal Saturado	$a = 0\ n < 0$ $a = n\ 0 \le n \le 1$ $a = 1\ n > 1$		*satlin*
Lineal Saturado Simétrico	$a = -1\ n < -1$ $a = n\ \ -1 \le n \le 1$ $a = +1\ n > 1$		*satlins*
Sigmoidal Logarítmico	$a = \frac{1}{1+e^{-n}}$		*logsig*
Tangente Sigmoidal Hiperbólica	$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		*tansig*
Competitiva	$a = 1$ *Neurona con n max* $a = 0$ *El resto de neuronas*		*compet*

Tabla 1. 2: Principales funciones de transferencia empleadas en el entrenamiento de Redes Neuronales Artificiales.

En última instancia queda la <u>función *salida*</u>, $f(.)$ Esta convierte el estado de la neurona en la salida hacia la siguiente neurona que se transmite por las sinapsis. Usualmente no se considera y se toma la identidad, de manera que la salida es el propio estado de activación de la neurona.

Existen algunas redes que transforman su estado de activación en una salida binaria y para eso usan la función escalón. Pero también existen otras posibilidades como, por ejemplo, las funciones probabilísticas.

1.3.5 Tipo de Aprendizaje.

Una de las capacidades que hacen atractivas a las Redes Neuronales Artificiales es su capacidad para aprender de su entorno y mejorar su respuesta de acuerdo con alguna medida predefinida a través de un proceso de aprendizaje, que se da a lo largo del tiempo.

Aprender significa básicamente que la red sufre un cambio de parámetros. Mientras dura el aprendizaje los parámetros libres de la red se adaptan a través de un proceso de estimulación del entorno en el que la red está inmersa (HAYKIN, 1994). El objetivo de dicho cambio es mejorar su respuesta al entorno que se cuantificará con la medida de que el algoritmo de aprendizaje dispone.

Existen diferentes tipos de aprendizaje que identifican diferentes maneras de relacionarse con el entorno:

- **Aprendizaje supervisado:** En este tipo de aprendizaje se le proporciona a la RNA una serie de ejemplos consistentes en unos patrones de entrada, junto con la salida que debería dar la red. El proceso de entrenamiento consiste en el ajuste de los pesos para que la salida de la red sea lo más parecida posible a la salida deseada. Es por ello que en cada iteración se usa alguna función que de cuenta del error o el grado de acierto que está cometiendo la red. Ejemplos de este tipo de redes son: el algoritmo de Aprendizaje por Cuantificación Vectorial (LVQ) de Kohonen, el Perceptrón Simple, la red Adaline, el Perceptrón Multicapa y la Memoria Asociativa Bidireccional.
- **Aprendizaje no supervisado o autoorganizado**: En este tipo de aprendizaje se presenta a la red una serie de ejemplos pero no se presenta la respuesta deseada. Lo que hace la RNA es reconocer regularidades en el conjunto de entradas. Ejemplos de este tipo de redes son: las memorias asociativas, las Redes de Hopfield, la Máquina de Bolzman y la Máquina de Cauchy, las redes de aprendizaje competitivo, los Mapas Autoorganizados(SOM) de Kohonen y las Redes de Resonancia Adaptativa (ART).

De manera general a la hora de clasificar en función del aprendizaje en las Redes Neuronales Artificiales, se suele hablar de los dos tipos de aprendizaje mencionados anteriormente, pero también se pueden ver dos más que seguidamente se presentan:

- **Aprendizaje reforzado:** Este aprendizaje descansa en la idea dual premio-castigo, donde se refuerza toda aquella acción que permita una mejora del modelo mediante la definición de una señal crítica. Un ejemplo es el algoritmo de Aprendizaje por Cuantificación Vectorial (LVQ) de Kohonen.
- **Redes híbridas:** Es una mezcla de los anteriores. Unas capas de la red tienen un aprendizaje supervisado y otras capas de la red tienen un aprendizaje de tipo no

supervisado. Son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de este último tipo son las redes de base radial.

1.3.6 Tipo de entrada

También se pueden clasificar las RNA según sean capaces de procesar información de distinto tipo en:

- **Redes analógicas:** procesan datos de entrada con valores continuos y, habitualmente, acotados. Ejemplos de este tipo de redes son: Hopfield, Kohonen y las redes de aprendizaje competitivo.
- **Redes discretas:** procesan datos de entrada de naturaleza discreta; habitualmente valores lógicos booleanos. Ejemplos de este segundo tipo de redes son: las Máquinas de Bolzman y Cauchy, y la red discreta de Hopfield.

1.4. Ventajas que ofrecen las Redes Neuronales Artificiales.

El interés por las Redes Neuronales Artificiales tiene su justificación en las fascinantes propiedades que estas poseen. Debido a su constitución y a sus fundamentos, 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 destacan:

- **Aprendizaje adaptativo**: capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.

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.

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.

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.

- **Auto-organización:** una RNA puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.

Las redes neuronales emplean su capacidad de aprendizaje adaptativo para auto-organizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto-organización 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 auto-organizan la información usada. Por ejemplo, la red llamada Back-Propagation, creará su propia representación característica, mediante la cual puede reconocer ciertos patrones.

- **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:** 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. Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.

Hay dos aspectos distintos respecto a la tolerancia a fallos:

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 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:** los cómputos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad. La estructura de una RNA es paralela, por lo cuál si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas 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 mínima.

- **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.

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.

- **Flexibilidad:** Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (ej. si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente).

1.5. Aplicaciones.

El campo de aplicación de las redes neuronales es prácticamente interminable. A continuación se presenta un pequeño resumen de estas:

Telefonía

Como ecualizador adaptable de canal. Este dispositivo, tuvo un gran éxito comercial, pues es una simple red neuronal, usada en sistemas de telefonía a larga distancia para estabilizar las señales de voz.

Aeroespacial

Creación de pilotos automáticos, de alta eficiencia en la conducción. Simulación de trayectorias de vuelo. Sistemas de control de aviones. Detección de fallas en los componentes del avión.

Automotriz

Sistema de guía automática de automóviles, análisis de garantía de automóviles.

Actividades bancarias

Chequeo y evaluación de tarjetas de crédito.

Defensa

Guía de proyectiles, seguimiento de objetivos, discriminación de objetivos, reconocimiento de rostros, censores, alarmas, radares, procesamiento de imágenes.

Electrónica

Predicción de secuencias de códigos, diseños de circuitos integrados, análisis de falla en Chips.

Finanzas

Apreciación del estado real de bienes, consejeras de préstamo, evaluación empresarial, análisis financiero, predicción de precios de monedas circulantes.

Industrial

Predicción de salida de gases en hornos y otros procesos industriales; además en control de procesos industriales, diseño y análisis de productos, diagnóstico de reparación de máquinas, identificación en tiempo real de partículas, sistemas de inspección visual de calidad, análisis de soldaduras, diseño y análisis de productos químicos, mantenimiento de máquinas.

Medicina

Análisis de células cancerígenas; diseño de prótesis, optimización de tiempos en trasplantes, reducción de gastos y acondicionamientos de hospitales.

Robótica

Control de trayectorias, manipulador de controles, sistemas de visión.

Telecomunicaciones

Compresión de imágenes y datos, servicios de información automatizada, traducción en tiempo real de voces a diferentes idiomas.

Transportación

Sistemas de diagnóstico del freno de un camión, sistemas de rutas, programación de viajes de vehículos.

1.6. Las Redes Neuronales aplicadas a la categorización de texto.

La Categorización de Texto (CT) consiste en clasificar un conjunto de documentos asignándole una o más categorías preexistentes a cada documento (LEWIS, 1992).

Los sistemas de categorización de texto necesitan un conjunto de documentos $\{d_1, d_2,\ldots d_n\}$, conocido con el nombre de colección de entrenamiento, etiquetados con categorías $\{c_1, c_2,\ldots c_m\}$. El objetivo de un sistema de categorización de texto consiste en decidir si un documento d_i pertenece o no a una categoría particular c_k. La mayoría de los sistemas de categorización dividen la colección de documentos en dos subconjuntos: un subconjunto con documentos de **entrenamiento** para predecir las categorías de nuevos documentos, y un subconjunto con documentos de **evaluación** que permitan comprobar la efectividad del sistema generado (LIU, 1999; YANG, 1999).

Para la categorización de texto se han propuesto una gran cantidad de algoritmos. La mayor parte de ellos no son, en realidad, específicos para clasificar documentos, sino que se han planteado para clasificar todo tipo de cosas. Sucede que algunos de éstos han sido utilizados (con más o menos adaptaciones) para la clasificación de documentos. Entre los más utilizados, están: algoritmos Probabilísticos, algoritmo de Rocchio, algoritmo del Vecino Más Próximo y variantes; y algoritmos basados en Redes Neuronales Artificiales. Existen varios trabajos que utilizan este último enfoque para desarrollar sistemas de categorización de texto. Un ejemplo es el de Park y Li (LI, 2006), donde se propone un algoritmo para categorizar textos que usa un Back-Propagation mejorado al que denominan MRBP (Morbidity Neuron Rectify Back-Propagation Neural Network) y se demuestra que funciona mejor que el normalmente usado (Back-Propagation Neural Network, BPNN), sobre todo cuando el tamaño de las redes es grande ya que elimina muchos de los defectos del BPNN y mantiene una regla ajustando las neuronas y entrenando la red eficazmente.

También Ruiz y Srinivasan (SRINIVASAN, 1998) presentan los resultados obtenidos de una serie de experimentos de categorización de texto de una colección de 2,344 artículos de MEDLINE[2]. Los experimentos comparan la actuación de una red Counter-Propagation contra una red Back-Propagation y demuestran que esta última tiene mejores resultados.

[2] MEDLINE es posiblemente la base de datos de bibliografía médica más amplia que existe. Producida por la Biblioteca Nacional de Medicina de los Estados Unidos. http://medline.cos.com/

Lam y Dominic Savio proponen un modelo usando Back-Propagation en la tesis: *"Learned Text Categorization By Back-Propagation Neural Network"* (SAVIO, 1996). Los resultados muestran que el modelo propuesto es efectivo en la categorización de texto y bastante preciso.

Goren-Bar, Kuflik y Lev (GOREN-BAR, 2000), entrenan redes neuronales que usan algoritmos de aprendizaje competitivo basados en el modelo de Kohonen en sus dos variantes: Mapa auto organizado (Self-Organizing Map) o SOM y Aprendizaje por Cuantificación Vectorial (Learning Vector Quantization) o LVQ. Usan un juego de documentos categorizados manualmente, extraídos de Internet (documentos sobre compañías y noticias financieras). Los resultados experimentales demuestran que el LVQ funciona mejor que el SOM.

Un trabajo que compara la efectividad de los distintos modelos de Kohonen para categorizar textos seleccionados de Internet es el de Rauber y Merkl (MERKL, 1999). Las pruebas se realizan con las dos versiones del modelo mostrando finalmente que el LVQ obtiene mejores resultados.

En Valdivia (VALDIVIA, 2004) se compara LVQ con el algoritmo de Rocchio, que ha sido el más utilizado en categorización de texto sirviendo de caso base para la comparación de otros sistemas desarrollados (MARTÍN-VALDIVIA, 2003; SEBASTIANI, 2002; UREÑA-LÓPEZ, 2002), se puede observar que los resultados obtenidos con el modelo LVQ están muy por encima de los del algoritmo de Rocchio. Otro ejemplo donde se obtienen magníficos resultados con el algoritmo LVQ es en el trabajo de Muhammad Fahad y Hayat Khiyal (KHIYAL, 2007) donde se realizan experimentos de categorización de texto utilizando la colección Reuters-21578[3]. Los resultados experimentales muestran que el LVQ mejora a otros métodos como Vecino Más Próximo (K-NN), Rocchio, Naïve Bayes (NB), Árboles de Decisión (Decision Tree) y Support Vector Machines (SVMs).

García Vega, Martín Valdivia y Ureña López en (MARTÍN-VALDIVIA, 2003), presentan un sistema de categorización usando la Biblia Políglota, en Español e Inglés, con un entrenamiento de textos multilingües. Comparan el ampliamente usado algoritmo de Rocchio con tres algoritmos que usan reglas de aprendizaje basadas en redes neuronales: Widrow-Hoff(WH) , Kivinen-Warmuth(KW) y LVQ. Finalmente demuestran que con el LVQ se obtienen óptimos resultados y que mejora bastante a los otros algoritmos.

[3] La colección de evaluación REUTERS-21578 para categorización de texto esta disponible libremente a través de Internet en: http://www.davidlewis.com/resources/testcollections/reuters21578/

Aunque no son muchos los trabajos que utilicen el algoritmo LVQ en categorización de texto, cabe señalar que los resultados obtenidos con este son realmente prometedores ya que fue desarrollado para ser un clasificador, sus características (que se abordarán detalladamente en el próximo capítulo) y el hecho de que se ha demostrado que funciona eficientemente en la categorización de texto constituyen motivos por los que se decide que esta tesis esté orientada a la utilización de este algoritmo.

1.7. Conclusiones.

En este capítulo se ha presentado una breve introducción a las redes neuronales, describiendo sus características básicas, clasificaciones y arquitectura así como algunas ventajas y aplicaciones. Finalmente se han comentando varios trabajos interesantes que manejan el enfoque de las redes neuronales aplicadas a la categorización de texto, haciendo énfasis en los algoritmos que utilizan. Se han resaltado las investigaciones que hacen uso del algoritmo LVQ puesto que será el que se utilice en esta tesis.

Capítulo 2 Representación de documentos y Modelo Neuronal de Kohonen.

En este capítulo se explica en qué consiste la clasificación asistida por computadora y de manera general como trabaja un clasificador. Luego se comenta como se procesan los documentos que se le darán como entrada al clasificador, haciendo un acercamiento a algunos de los mecanismos que se usan para esta tarea, más específicamente a los que se utilizaran en este trabajo. También se presenta como funciona el modelo de espacio vectorial, seleccionado para la representación de la información.

Otro aspecto que se define es el modelo neuronal escogido para lo cual se da una breve introducción al tipo de aprendizaje que este utiliza. Además se especifican las métricas que se emplean para evaluar la red.

2.1. Clasificación asistida por computadora.

La decisión de clasificar objetos en un número predefinido de categorías está basada en las características o propiedades de los objetos. Para llevar a cabo la tarea de clasificación, una representación eficaz de los objetos debe incluir todas las propiedades que son útiles para identificar la categoría a la que los objetos deben asignarse (SAVIO, 1996).

En la clasificación asistida por computadora, el objeto es usualmente representado por un juego de atributos o rasgos. Cada rasgo corresponde a una propiedad del objeto a ser representado. Por lo general este juego de rasgos es agrupado para formar un vector, en el que cada componente del vector corresponde a un rasgo. Por ejemplo el siguiente es un vector (X) con n rasgos $(f_1, f_2, \ldots f_n)$:

$$X =< f_1, f_2, \ldots f_n > \qquad (2.1)$$

El tipo de datos del vector depende del tipo de rasgos que se representan.

En la figura 2.1 se muestra un modelo general de la tarea de clasificación asistida por computadora, en este modelo, un juego de vectores de rasgo n-dimensional que representan los

objetos a ser clasificados se da como entrada al clasificador. Basado en los rasgos contenidos en cada vector de entrada, el clasificador tomara la decisión de asignar una determinada categoría para los objetos.

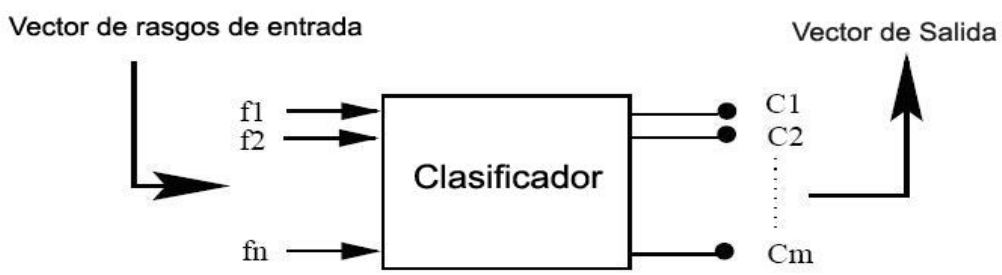

Figura 2. 1: Modelo general de clasificación asistida por computadora.

Esta decisión es representada por un vector de salida m-dimensional donde m es igual al número de categorías predefinidas en la tarea de clasificación particular. En este caso, el clasificador puede verse como una función que mapea un vector de rasgo n-dimensional en un vector de salida m-dimensional. Al vector de salida se le suele llamar vector de clasificación.

Dependiendo de la tarea de clasificación el vector de salida puede ser un vector binario o un vector con valores reales. En la clasificación binaria, el número de miembros de la categoría de un objeto es binario. Es decir una categoría es asignada o no a un objeto. Un uno en un determinado componente del vector de salida indica que la categoría es asignada al objeto, mientras que un cero significa que la categoría no es asignada al objeto. Si existen solapamientos entre categorías, donde un objeto puede ser asignado a más de una categoría, dos o más bits en el vector de salida estarán en uno.

Por otra parte en la clasificación probabilística, el vector de salida será un valor real, donde cada componente del vector de salida será un número real, usualmente entre el rango de $[0,1]$. El valor de un determinado componente del vector representa la probabilidad del objeto de pertenecer a una determinada categoría. Un mayor valor en un componente determinado del vector indica mayor probabilidad de que el objeto sea miembro de la categoría, mientras que un menor valor indicaría todo lo contrario.

2.2. Selección de los rasgos

La selección del juego de rasgos para describir los objetos a ser clasificados puede tener un gran impacto en la exactitud del sistema de clasificación. Es por eso que para incrementar la exactitud del clasificador, es bueno tener una buena representación de los rasgos del objeto, que es considerado un paso crítico y que consume más tiempo en la construcción de un sistema clasificador (SANZ, 2004).

El proceso de análisis que obtiene la representación de un documento se denomina *indexación*. Se suelen utilizar numerosas técnicas para mejorar dicho proceso con el fin de aumentar el rendimiento de los sistemas. Para ello se deben extraer un conjunto de términos adecuados, que por ser muy elevado se hace necesario realizar algún proceso de reducción. Existen algunos mecanismos utilizados para esto, entre los que se encuentran: la semántica latente, el análisis de componentes principales, las listas de parada, la extracción de raíces, etc. Seguidamente se comentan los dos últimos, pues serán los que se utilicen en este trabajo.

2.2.1 Listas de paradas.

En los documentos aparecen términos muy frecuentes y con poco contenido semántico como, por ejemplo, las palabras en inglés (*"the", "a", "and", "that",…*), que están presentes en muchísimos documentos y no aportan prácticamente información del tema tratado. Estos términos no son útiles en las tareas de categorización de texto, se denominan *palabras vacías* (*stop-words*) y es conveniente su eliminación en el proceso de indexado. Para ello se crea una lista de parada (*stoplist*) y luego se filtra de modo que cada palabra que aparezca en la lista de parada es eliminada del documento.

El mayor problema está en decidir que palabras deben ser añadidas a la lista de parada. Esto depende muchas veces del tema que trate el documento porque existen palabras que puede que no sean consideradas palabras vacías en un documento en inglés de manera general y sin embargo puede que no sea así en una colección de documentos especializados. Un ejemplo de ello es ("home", "page", "world", "web") que no son consideradas palabras vacías en general en el inglés y sin embargo en una colección de documentos relacionados con la World Wide Web (WWW) si pueden ser consideradas como tal. En el caso de que los temas de los documentos

sean desconocidos durante la construcción del sistema, las listas de paradas que se escogen son usualmente las del lenguaje que se desea categorizar en general.

Las listas de parada se obtienen mediante estudios orientados específicamente a ello, a partir de un corpus de texto lo suficientemente representativo del idioma considerado. La comunidad científica dispone de listas de palabras vacías para numerosos idiomas, entre las que se incluyen también algunos verbos, adverbios o adjetivos de uso frecuente.

2.2.2 Extracción de raíces.

Existen palabras en un documento que son variaciones morfológicas, que aunque aparecen en diferentes formas, representan el mismo concepto como, por ejemplo, limpio, limpia, limpios, limpias.

Los algoritmos de extracción de raíces tienen como objetivo obtener un único término de indexación a partir de las diferentes variaciones morfológicas de una palabra, como por ejemplo: “comunicación”, “comunicante” o “comunicado”, compartirían la raíz “comunic-”, que no será necesariamente la raíz lingüística de la palabra.

Dentro de estas técnicas, los stemmers más conocidos son los algoritmos de Lovins (LOVINS, 1968), Dawson (DAWSON, 1974), Porter (PORTER, 1980), y Paice (PAICE, 1990) y se suelen aplicar al idioma inglés (GÁLVEZ, 2006). Entre estos métodos uno de los más utilizados es el algoritmo de Porter que permite extraer los sufijos y prefijos comunes de palabras literalmente diferentes pero con una raíz común que pueden ser consideradas como un sólo término, asegurando que la forma de las palabras no penalice la frecuencia de estas.

2.3. Modelo de representación de información.

El procesamiento de los documentos consta de dos etapas fundamentales: la primera etapa es el preprocesado de los documentos, que consiste en preparar los documentos para su parametrización, eliminando aquellos elementos que se consideran superfluos. Aquí se lleva a cabo un proceso de indexación y usualmente se utilizan mecanismos como los que han sido mencionados anteriormente: las listas de paradas, eliminando las palabras vacías y la extracción de raíces, transformando los términos en la raíz mediante algún algoritmo de stemming. La otra

etapa es la parametrización que es una etapa de complejidad mínima una vez se han identificado los términos relevantes. Consiste en realizar una cuantificación de las características (es decir, de los términos) de los documentos; que no es más que construir un vector para cada documento con tantas componentes como términos han quedado en la lista. Esto se puede realizar de varias formas, los modelos más utilizados son el modelo booleano, el modelo probabilístico y el modelo de espacio vectorial. A continuación se comenta este último que será el que se utilice en esta tesis.

2.3.1 Modelo de espacio vectorial.

En el modelo de espacio vectorial o MEV (Space Vector Model-SVM) (SALTON, 1975) se considera el carácter semántico de los documentos, mediante la asignación de pesos a los términos, que indica su presencia o importancia en el documento o en la colección.

La idea básica de este modelo reside en la construcción de una matriz de términos y documentos, donde las columnas fueran estos últimos y las filas correspondieran a los términos incluidos en ellos. Así, las columnas de esta matriz (que en términos algebraicos se denominan vectores) serían equivalentes a los documentos que se expresarían en función de las apariciones (frecuencia) de cada término. De esta manera, un documento podría expresarse de la manera d1= (1, 2, 0, 0, 0,... ... ..., 1, 3) siendo cada uno de estos valores el número de veces que aparece cada término en el documento. La longitud del vector de documentos sería igual al total de términos de la matriz (el número de filas).

De esta manera, un conjunto de m documentos y de n términos almacenados en ese conjunto de documentos sería una matriz de n filas por m columnas. El valor de cada componente del vector representa la importancia relativa de ese término en el documento. Una aproximación común para el pesado de términos usa la frecuencia de ocurrencia de una palabra determinada en el documento para representar las componentes del vector. Para calcular los pesos de los términos se usa la ecuación estándar $tf \cdot idf$, donde tf (*term frequency*) identifica la frecuencia del término en el documento, y idf (*inverse document frecuency*) es la inversa de la frecuencia de documento definida como:

$$idf_i = \log_2 (M / df_i) \qquad (2.2)$$

Donde df_i (document frequency-frecuencia del documento) identifica al número de documentos de la colección en el que se encuentra presente el término i y M es el número total de documentos en la colección.

De esta manera, el peso del término i en el documento $d_j (w_{ij})$ se calcula según la siguiente ecuación:

$$w_{ij} = tf_{ij} \cdot idf_i \tag{2.3}$$

donde tf_{ij} es el número de ocurrencias del término i en el documento d_j.

Así, cada documento d_j queda representado como un vector de la forma:

$$d_j = (w_{1j}, w_{2j}, ..., w_{nj}) \tag{2.4}$$

En tareas en las que las clases están predefinidas (como por ejemplo, en la CT) es preciso obtener una representación de las categorías. Para ello se utilizan vectores de pesos con los términos que se incluyen en cada categoría.

Cada categoría c_k queda representada como un vector de la forma:

$$c_k = (c_{1k}, c_{2k}, ... c_{nk}) \tag{2.5}$$

La similitud entre el documento d_j y la categoría c_k se puede calcular de diferentes formas, ejemplos son: la medida del coseno del ángulo entre los vectores, la distancia euclídea, el producto escalar de dos vectores, etc, siendo la primera una de las más utilizadas. Así, la similitud entre el documento d_j y la categoría c_k se obtiene según la siguiente ecuación:

$$sim(d_j, c_k) = \frac{\sum_{i=1}^{N} w_{ij} \cdot c_{ik}}{\sqrt{\sum_{i=1}^{N} w^2{}_{ij} \cdot \sum_{i=1}^{N} c^2{}_{ik}}} \tag{2.6}$$

2.4. Aprendizaje.

Una cuestión importante en un sistema clasificador de texto es el algoritmo de aprendizaje que emplea el modelo seleccionado para la clasificación. En este caso se ha escogido la versión original del modelo Kohonen: algoritmo de Aprendizaje por Cuantificación Vectorial (Learning Vector Quantization-LVQ). Para una mejor comprensión del modelo se explicará primeramente en que consiste el aprendizaje competitivo, que es el que este utiliza.

2.4.1 Aprendizaje competitivo.

En las redes con aprendizaje competitivo suele decirse que las neuronas compiten entre sí con el fin de llevar a cabo una tarea determinada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active (alcance su valor de respuesta máximo). Por tanto las neuronas compiten para activarse quedando finalmente una, o una por grupo, como neurona vencedora y el resto quedan anuladas y son forzadas a sus valores de respuestas mínimos. Esto se conoce con el nombre de *Winner-Take-All* (WTA).

En las redes competitivas existen neuronas con conexiones de autoexitación (signo positivo) y conexiones de inhibición (signo negativo) por parte de neuronas vecinas (conexiones laterales). En la figura 2.2 se pueden observar estas conexiones. Aquí se representan las conexiones de una sola unidad.

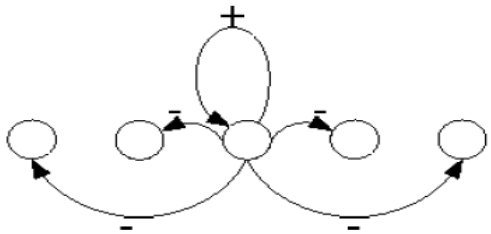

Figura 2. 2: Conexiones autoexitatorias e inhibitorias de una capa competitiva.

El objetivo de este aprendizaje es categorizar los datos que se introducen en la red, de esta forma las informaciones similares son clasificadas formando parte de la misma categoría. Así, dichos

valores de entrada deberán activar la misma neurona de salida que será la que represente a la categoría a la que pertenecen estos datos.

Si el aprendizaje es no supervisado, las categorías deben ser creadas por la misma red a través de las correlaciones entre los datos. En caso contrario, las clases vendrán determinadas por los datos de entrenamiento.

Tuevo Kohonen ha sido un fuerte proponente de las redes competitivas, sin embargo su énfasis ha sido en aplicaciones para ingeniería y en descripciones de eficiencia matemática de las redes.

En 1982 Kohonen se basa en los trabajos de Grossberg y Von der Malsburg para desarrollar un modelo que permite incorporar topología a una red competitiva utilizando el principio de inhibición lateral[4]. Su principal aportación consiste en un procedimiento para conseguir que unidades físicamente adyacentes aprendieran a representar patrones de entrada similares.

2.5. Modelo neuronal de Kohonen.

Existen evidencias que demuestran que en el cerebro hay neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales.

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje, esto sugiere que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior, de hecho esta teoría podría explicar su poder de operar con elementos semánticos: algunas áreas del cerebro simplemente podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones, en definitiva se construirían mapas especiales para atributos y características.

A partir de estas ideas Tuevo Kohonen presentó en 1982 un sistema con un comportamiento semejante, se trataba de un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro; el objetivo de Kohonen era

[4] La inhibición lateral ocurre cuando una neurona se activa, ésta produce un estado excitatorio en las células más cercanas y un efecto inhibitorio en las más lejanas.

demostrar que un estímulo externo (información de entrada) por si solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas.

Este modelo tiene dos variantes denominadas LVQ (Learning Vector Quantization) y TPM (Topology Preserving Map) o SOM (Self Organizing Map), ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de estos, siendo de una sola dimensión en el caso de LVQ y bidimensional o tridimensional en la red SOM.

El aprendizaje en el modelo de Kohonen es de tipo Off-line, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. En la etapa de aprendizaje se fijan los valores de las conexiones (feedforward) entre la capa de entrada y la salida. En ambos modelos se utiliza un aprendizaje competitivo por refuerzo. Puesto que se trata de un aprendizaje por refuerzo, si la neurona acierta en la salida, se le premia modificando sus pesos de conexión positivamente de manera que se refuerza dicha conexión. Si por el contrario, la neurona falla, sus pesos se modifican negativamente como castigo al error cometido.

Durante la etapa de entrenamiento, se presenta a la red un conjunto de informaciones de entradas (vectores de entrenamiento) para que esta establezca en función de la semejanza entre los datos las diferentes categorías (una por neurona de salida), que servirían 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. En el caso de existir más patrones de entrenamiento que neuronas de salida, más de una deberá asociarse con la misma neurona, es decir pertenecerán a la misma clase.

En este modelo 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.

A continuación se explica con mayor profundidad una de las variantes del modelo de Kohonen: LVQ que como se ha dicho antes será el algoritmo de aprendizaje que se utilice en este trabajo.

2.5.1 Aprendizaje por Cuantificación Vectorial.

El algoritmo de Aprendizaje por Cuantificación Vectorial o LVQ se puede ver como la versión supervisada del modelo de Kohonen. Utiliza también un aprendizaje competitivo y por refuerzo, pero en este caso los datos de entrenamiento contienen la respuesta correcta para cada modelo de entrada. En cada etapa del entrenamiento solamente la unidad que gana la competición será la que modificara sus pesos de conexión.

2.5.1.1 Arquitectura y entrenamiento de la red.

En cuanto a la arquitectura de red, el modelo LVQ utiliza una red de dos capas. Sus conexiones se realizan hacia delante (feedforward). Los pesos de conexión se representan mediante una matriz W^1 de $S^1 \times R$ pesos (Figura 2.3).

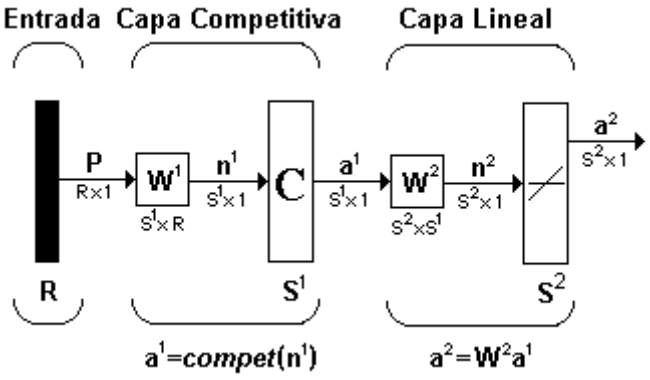

Figura 2. 3: Red LVQ.

En este modelo cada neurona de la primera capa es asignada a una clase, después cada clase es asignada a una neurona en la segunda capa. El número de neuronas en la primera capa, S^1 debe ser mayor o al menos igual que el número de neuronas en la segunda capa S^2.

El modelo de Kohonen debe ser ajustado mediante una primera fase de entrenamiento, para posteriormente ser utilizado en su fase de producción.

El proceso de entrenamiento empieza con la inicialización de los pesos sinápticos W^1. La inicialización se puede realizar de varias formas, entre ellas están: pesos nulos, aleatorios de pequeños valor absoluto, o con un valor de partida predeterminado. Luego se escoge aleatoriamente un vector p del conjunto de entrada R. Cada vector prototipo ${}_i w^1$, que son los vectores que representan los pesos de conexión, calcula su distancia con respecto al vector de entrada en base a alguna función (producto interno, función coseno, distancia Hamming…). Un criterio de medida muy utilizado es la distancia euclídea:

$$\left\| R - {}_i w^1 \right\| = \sqrt{\sum_j (r_j - W^1)^2} \qquad (2.7)$$

Después se establece una competición entre los vectores prototipos y el vector de entrada para determinar cuál es el más cercano al vector de entrenamiento que será proclamado ganador de dicha competición. Es decir se determina la neurona ganadora, cuya distancia sea la menor de todas.

La salida de la primera capa de la red LVQ sería:

$$a^1 = compet(n^1) \qquad (2.8)$$

Donde a es la salida y n es la entrada neta a la función de transferencia, esta entrada neta seria:

$$\boldsymbol{n}_i^1 = -\begin{bmatrix} \left\| {}_1\boldsymbol{w}^1 - \boldsymbol{p} \right\| \\ \left\| {}_2\boldsymbol{w}^1 - \boldsymbol{p} \right\| \\ \vdots \\ \left\| {}_{S^1}\boldsymbol{w}^1 - \boldsymbol{p} \right\| \end{bmatrix} \qquad (2.9)$$

$Compet$ es una función de transferencia que encuentra el índice i^* de la neurona con la entrada neta más grande y fija su salida en uno, el resto de las neuronas tienen salida 0.

Entonces la neurona cuyo vector de pesos esté cercano al vector de entrada tendrá salida 1 y las otras neuronas, tendrán salida 0; la salida no cero representa una sub- clase, muchas neuronas (subclases), conforman una clase (Figura 2.4).

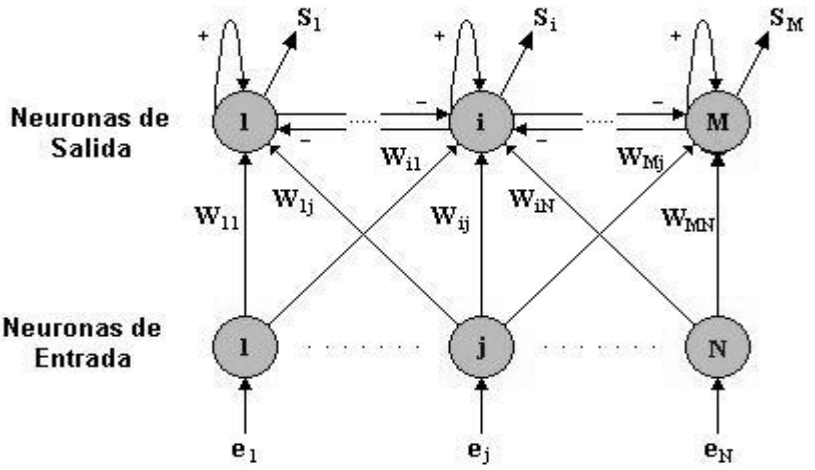

Figura 2. 4: Comportamiento de las neuronas en una red LVQ.

La segunda capa de la red LVQ es usada para combinar subclases dentro de una sola clase, esto es realizado por la matriz de pesos W^2. Las columnas de W^2 representan las subclases y las filas representan las clases, W^2 tiene un solo 1 en cada columna, todos los demás elementos son cero, la fila en la cual se presenta el 1 índica cuál es la clase a la que la subclase pertenece.

$$W^2{}_{ki} = 1 \quad \text{; la subclase } i \text{ pertenece a la clase } k \tag{2.9}$$

Una propiedad importante de esta red, es que el proceso de combinar subclases para formar clases, permite la creación de clases más complejas. Una capa competitiva estándar tiene la limitación de que puede crear solo regiones de decisión convexas; la red LVQ soluciona esta limitación.

Para generar la matriz W^2, antes de que suceda el aprendizaje cada neurona en la segunda capa es asignada a una neurona de salida; por lo general igual número de neuronas ocultas son conectadas a cada neurona de salida, para que cada clase pueda ser conformada por el mismo número de regiones convexas.

Una vez que W^2 ha sido definida nunca será alterada.

Para obtener la salida final se multiplica $W^2 a^1$ que indicará a quien está siendo asignado el vector de entrada p presentado inicialmente a la red.

Se procede entonces a la actualización de los pesos sinápticos de la neurona ganadora por medio de la regla de Kohonen que es empleada para mejorar la capa oculta de la red LVQ.

Puesto que los vectores prototipos y los patrones de entradas están etiquetados con las clases a las que pertenecen, las correcciones son del tipo premio o castigo. Si se acertó en la clasificación (la clase del vector prototipo y la clase del vector de entrada coinciden) se premia al vector prototipo acercándolo al vector de entrada. Por el contrario, si la clasificación no se ha realizado correctamente (la clase seleccionada por el vector prototipo es diferente a la clase del vector de entrada), se castiga al vector prototipo alejándolo del vector de entrada. La modificación de los pesos en el instante t se realiza según la siguiente ecuación:

$$ {}_i w^1(t+1) = \begin{cases} {}_i w^1(t) + \alpha(t)[R(t) - {}_i w^1(t)] \\ \qquad \text{si } {}_i w^1(t) \text{ y } R(t) \text{ pertenecena la misma clase} \\ {}_i w^1(t) - \alpha(t)[R(t) - {}_i w^1(t)] \\ \qquad \text{si } {}_i w^1(t) \text{ y } R(t) \text{ pertenecena clases diferentes} \\ {}_i w^1(t) \quad \text{si } i \neq k \end{cases} \tag{2.10} $$

Donde $\alpha(t)$ es la *tasa de aprendizaje*. La *tasa de aprendizaje* es un parámetro que utiliza muchas redes neuronales en su fase de entrenamiento para controlar la velocidad de convergencia de los pesos sinápticos. Resulta de gran importancia la elección de un valor adecuado para la *tasa de aprendizaje* que permita hallar un punto de equilibrio entre la velocidad de convergencia y la estabilidad final de los vectores prototipo. Una *tasa de aprendizaje* cercana a 0, hace que el aprendizaje sea lento pero asegura que cuando un vector prototipo haya alcanzado el centro de una clase, se mantenga allí indefinidamente. Por el contrario, valores altos de $\alpha(t)$ cercanos a 1 hacen que el acercamiento del vector prototipo al vector de entrada sea muy rápido, pero los pesos tendrán grandes oscilaciones provocando, en ocasiones, cierta inestabilidad. Se suele inicializar $\alpha(t)$ a algún valor no muy alto, como por ejemplo 0,3; y se decrementa a medida que avanza el entrenamiento según alguna función, de manera que al final del proceso su valor sea

muy cercano a 0. Normalmente, por motivos computacionales, se suele elegir una función lineal, monótona decreciente y que tome valores en el intervalo (0,1), como la mostrada a continuación:

$$\alpha(q) = \alpha_1(1 - \frac{q}{\alpha_2}) \qquad (2.11)$$

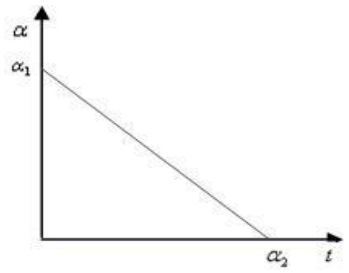

Figura 2. 5: Representación gráfica de la ecuación lineal decreciente (2.11).

El proceso de entrenamiento que sigue el algoritmo LVQ se muestra de manera resumida en la siguiente secuencia de pasos:

1. Inicializar los pesos de conexión W^1.
2. Se presenta a la red un vector de entrada $p = (p_1, p_2, ... p_n)$ seleccionado aleatoriamente y se calcula la distancia a cada vector prototipo.
3. Las neuronas ocultas compiten, la neurona i^* gana la competición y el i^*-ésimo elemento de a^1 se fija en 1 (ecuación 2.8).
4. a^1 es multiplicada por W^2 para obtener la salida final a^2, la cual tiene solamente un elemento no cero, k^*, indicando que el patrón p está siendo asignado a la clase k^*.
5. Modificar los pesos de la unidad ganadora premiándola o castigándola según sea el caso (ecuación 2.10)
6. Decrementar $\alpha(t)$.
7. Repetir desde el paso 2 hasta que los pesos no cambien o durante un número fijo de iteraciones.

Uno de los problemas que presenta el algoritmo LVQ esta relacionado con la estabilidad de la red que se produce cuando las clases están muy juntas, ocurriendo en ciertos casos que un vector de

pesos que intenta apuntar a una determinada clase, acabe entrando en la región de otro vector de pesos cercano. Otro problema que posee este algoritmo está relacionado con los pesos de conexión y se conoce con el nombre de *neurona muerta*. Es posible que inicialmente el vector de pesos asociado a una neurona de salida se encuentre muy lejos de cualquiera de los vectores de entrada y, por lo tanto, nunca sea capaz de ganar la competición, con lo cual, nunca se le permitirá aprender o sea modificar sus pesos. Se trata de una neurona muerta en la red que debería ser eliminada o forzada a aprender. Una posible solución consiste en añadir una ganancia negativa a la entrada de cada neurona y decrementar así la ganancia total cada vez que la neurona gane la competición. Con esto, se hace más difícil que una neurona gane sucesivamente la competición. Este mecanismo se conoce con el nombre de *conciencia*.

Existen variantes del algoritmo LVQ. La versión original del modelo propuesto por Kohonen se conoce como LVQ1 y es la que se ha descrito hasta aquí. Posteriormente Kohonen y sus colaboradores proponen varias modificaciones del algoritmo conocidas como LVQ2.1, LVQ3 y OLVQ1.

Es difícil decidir cuál sería la versión más apropiada para un problema dado. Muchos autores incluido Kohonen sostienen que los resultados obtenidos con las distintas estrategias son similares. Por ello, mayormente se suele utilizar el LVQ1 ya que es el que requiere el ajuste de menos parámetros. Por esto en esta tesis se realizan los experimentos con el algoritmo LVQ1 originalmente propuesto por Kohonen y que se puede llamar simplemente LVQ.

2.6. Métricas de evaluación.

Una cuestión importante en el desarrollo de sistemas es la evaluación del mismo. Aunque un sistema se debe medir en términos de eficiencia, que es la que se ocupa de aspectos tales como el tiempo de ejecución, el espacio de almacenamiento…etc; y efectividad, que es la que se orienta a temas de calidad de resultados; este trabajo se centra principalmente en la evaluación de la efectividad.

Existen diferentes medidas que se han propuesto para medir la efectividad de un sistema clasificador. Dos medidas ampliamente utilizadas son: la *precisión* y la *tasa de recuperación* o *recall*. Estas inicialmente se aplicaron en el ámbito de la recuperación de información pero

posteriormente se extendió su uso a otras tareas ajustando en cada caso cada una de las medidas.

La precisión puede ser vista como una medida de exactitud o fidelidad, mientras que el *recall* es una medida de integridad.

Precisión

$$P = \frac{\text{elementos correctamente clasificados como pertenecientes a la categoria}}{\text{total de elementos clasificados como pertenecientes a la categoria}} \quad (2.11)$$

Recall

$$R = \frac{\text{elementos correctamente clasificados como pertenecientes a la categoria}}{\text{total de elementos que en realidad pertenecen a la categoria}} \quad (2.12)$$

Así la precisión representa la capacidad del clasificador para evitar clasificar en una categoría aquellos documentos que no deben ser categorizados con esa categoría (ecuación 2.11) y el *recall* representa aquellos documentos que deberían ser clasificados en una determinada categoría (ecuación 2.12).

De esta manera una tabla de contingencia para cada categoría sería:

	Si es correcto	**No es correcto**
El sistema dice Si	***tp***	***fp***
El sistema dice No	***fn***	***tn***

Tabla 2. 1: Tabla de contingencia.

A continuación se calcula la precisión y el *recall* según las ecuaciones 2.13 y 2.14.

$$P = \frac{tp}{tp + fp} \quad (2.13)$$

$$R = \frac{tp}{tp + fn} \tag{2.14}$$

Donde tp son los verdaderos positivos; fp son los falsos positivos y fn son los falsos negativos.

Se puede medir la efectividad media de un clasificador de una manera global mediante otras dos medidas: la precisión *microaveraging* P_{μ} (ecuación 2.15) que consiste en calcular la precisión media para todas las categorías y la precisión *macroaveraging* P_{macro} (ecuación 2.16) que consiste en calcular la media de la precisión de cada una de las categorías.

$$P_{\mu} = \frac{\sum_{i=1}^{K} tp}{\sum_{i=1}^{K} (tp + fn)} \tag{2.15}$$

$$P_{macro} = \frac{\sum_{i=1}^{K} P}{K} \tag{2.16}$$

Donde *K* es el número de categorías.

En algunas ocasiones se suelen combinar la *precisión* y el *recall* para dar lugar a otras medidas que permitan comprobar la efectividad del sistema con un único valor. Siendo una de las medidas más utilizadas la conocida como la medida F_1:

$$F(R,P) = \frac{2PR}{P+R} \tag{2.17}$$

2.7. Conclusiones.

En este capítulo se ha presentado primeramente como funciona un clasificador de texto de manera general. Se han destacado los pasos principales que se llevan a cabo para procesar los documentos que se le dan como entrada al clasificador, haciendo énfasis en las listas de parada y los algoritmos de extracción de raíces, que son los mecanismos para reducir los vectores de

rasgos, que se aplicaran a los experimentos que se realicen en este trabajo. Se ha explicado el Modelo de espacio vectorial que es el que se utilizará posteriormente para representar la información.

Seguidamente se han dado los fundamentos del aprendizaje competitivo que es una de las bases para entender el modelo neuronal que se propone. Luego se ha estudiado el modelo neuronal de Kohonen analizando en detalle su versión supervisada, el algoritmo LVQ. De este se ha presentado su arquitectura y su algoritmo de entrenamiento, así como algunas características que resultan verdaderamente interesantes.

Finalmente se ha dedicado un apartado a las métricas recalcando las que servirán para evaluar el sistema.

Capítulo 3 Algoritmo LVQ aplicado a la categorización de texto.

En este capítulo se estudia la aplicación del algoritmo LVQ a la categorización de texto. Primeramente se comenta de manera general la clasificación automática de documentos. A continuación se presentan los experimentos realizados con el algoritmo LVQ. Para ello se describe la colección de entrenamiento utilizada, de manera resumida la herramienta empleada para realizar el entrenamiento y las pruebas; y finalmente se realiza una evaluación y se analizan los resultados obtenidos.

3.1. Clasificación automática de documentos.

Por clasificación automática de documentos se entiende, en sentido amplio, un conjunto de algoritmos, técnicas y sistemas capaces de asignar un documento a una o más clases, construidos según su afinidad temática. Básicamente, se tienen dos escenarios posibles: cuando el propio sistema debe determinar qué clases o grupos han de producirse; y cuando las clases o grupos son determinados a priori por personas. En este último caso se suele hablar de categorización o clasificación automática supervisada, ya que requiere supervisión o intervención de personas, tanto para diseñar las clases o categorías, como para enseñar o entrenar al sistema. Para esto se utilizan colecciones de entrenamiento, cuyos documentos han sido previamente categorizados.

La clasificación automática de documentos ha sido ampliamente estudiada por diversos investigadores. Puede contemplarse como un proceso de “aprendizaje matemático-estadístico”, durante el cual un algoritmo implementado computacionalmente capta las características que distinguen cada categoría o clase de las demás, es decir, aquellas que deben poseer los documentos para pertenecer a esa categoría. Estas características no tienen por qué indicar de forma absoluta la pertenencia a una clase o categoría, sino que más bien lo hacen en función de una escala o graduación. De esta forma, por ejemplo, documentos que posean una cierta característica tendrán un factor de posibilidades de pertenecer a determinada clase. De modo que la acumulación de dichas cantidades puede arrojar un resultado consistente en un coeficiente asociado a cada una de las clases existentes. Este coeficiente lo que expresa en realidad es el grado de confianza o certeza de que el documento en cuestión pertenezca a la clase asociada al coeficiente resultante.

Como se decía anteriormente los sistemas de categorización de texto necesitan un conjunto de documentos etiquetados con categorías que se conoce con el nombre de colección y que mayormente se divide en dos subconjuntos [Yang, 1999; Yang y Liu, 1999]:

- Un subconjunto con documentos de entrenamiento que permite predecir las categorías de nuevos documentos.
- Un subconjunto con documentos de evaluación que permita comprobar la efectividad del sistema generado.

A continuación se describe en detalle el experimento realizado para categorizar texto con el algoritmo LVQ.

3.2. Descripción de los experimentos.

Para cumplir el objetivo de esta tesis se construyó una colección de entrenamiento manualmente que está compuesta por documentos de noticias de prensa, extraídas de los periódicos: Granma[5], Juventud Rebelde[6], Agencia Cubana de noticias (ACN)[7], Prensa Latina[8], Cubarte[9], CNN[10]; y de un sitio que publica cuentos eróticos[11]. Todos estos documentos están en el idioma inglés. Se escogió este lenguaje debido a que es el que más está presente en Internet según estudios realizados por Dirección Terminología e Industrias de la Lengua [DTIL 2007]. En las tablas 3.1 y 3.2 se muestran algunas estadísticas.

[5] http://www.granma.cu/ingles/index.html
[6] http://www.juventudrebelde.co.cu/
[7] http://www.cubanews.ain.cu/
[8] http://www.prensa-latinaenglish.com/
[9] http://www.cubarte-english.cult.cu/
[10] http://edition.cnn.com/
[11] http://stories.xnxx.com/story/

Porcentaje de paginas WEB Mayo 2007	
Inglés	45,00%
Español	3,80%
Francés	4,41%
Italiano	2,66%
Portugués	1,39%
Rumano	0,28%
Alemán	5,90%
Catalán	0,14%
Resto	36,54%

Tabla 3. 1: Porcentajes totales de páginas Web en los idiomas del estudio.

	Inglés		Español		Portugués		Francés		Italiano		Rumano		Alemán		Catalán		Otros	
	2005	2007	2005	2007	2005	2007	2005	2007	2005	2007	2005	2007	2005	2007	2005	2007	2005	2007
Millones de internautas	295,4	366	72,0	101,5	24,4	47,3	33,9	58,4	30,4	31,4	4,4	4,9	55,3	58,9	X	2,1	313,16	483,7
% de internautas	35,62	31,7	8,68	8,8	2,94	4,1	4,09	5,1	3,67	2,7	0,53	0,4	6,67	5,1	X	0,2	37,81	41,9

Tabla 3. 2: Número de internautas por lengua.

3.2.1 La colección de entrenamiento.

Se definieron cinco categorías, cuatro adecuadas y una nociva: cultura, ciencia y tecnología, deporte, economía y erótico respectivamente. Estas categorías fueron escogidas de este modo debido a que el proyecto FILPACON tiene como objetivo fundamental regular el acceso a Internet partiendo de algoritmos inteligentes que categoricen las páginas Web de acuerdo a su contenido. Se han adquirido 200 documentos por cada categoría, por lo que la colección de entrenamiento que se ha utilizado está compuesta por 1000 documentos.

Puede notarse que aunque se han dividido los documentos en categorías, entre algunas de ellas pudieran darse solapamientos: por ejemplo, entre Tecnología y Economía, ya que avances tecnológicos pueden traer repercusiones satisfactorias en la economía.

3.2.2 Preprocesado de los documentos.

Para poder llevar a cabo la clasificación automática es preciso primeramente procesar los documentos. Para ello se convirtieron todas las mayúsculas a minúsculas y se eliminaron los números. Se aplicó un filtro para reducir la cantidad de letras consecutivas repetidas más de tres veces en una palabra. Se han eliminado las palabras vacías que se encuentran en la lista de parada del sistema SMART[12], que contiene un total de 571 palabras, de modo que cada término que aparece en la lista de parada ha sido eliminado de los documentos. También se eliminaron las 1000 palabras más frecuentes del idioma inglés[13] debido a que estas palabras tienen una gran probabilidad de aparecer en casi todos los documentos y haría que el *idf* (*inverse document frecuency*) fuera cero y entonces la representación del término en el espacio vectorial sería cero. Esta lista de las 1000 palabras más frecuentes del inglés tiene un total de palabras procesadas de: 66406121, con un total de palabras diferentes de: 240958; y las palabras más comunes aparecen en orden de acuerdo a su frecuencia de aparición. En la figura 3.1 se muestra un fragmento de esta lista.

```
# words : 66406121
# distinct words : 240958
mean frequency : 275.38    (standard deviation: 11650.4)
rank: 1 the     3837469
rank: 2 of      1644414
rank: 3 to      1642834
rank: 4 a       1636088
rank: 5 and     1523004
```

Figura 3. 1: Fragmento de la lista de palabras más frecuentes del inglés.

[12] SMART es uno de los principales sistemas de recuperación de información de dominio publico disponible a través de Internet en la dirección ftp://ftp.cs.cornell.edu/pub/smart

[13] Las 1000 palabras más frecuentes del idioma inglés están disponibles en: http://members.unine.ch/jacques.savoy/clef/index.html

Luego, a cada término se le aplicó la función de stemming de Porter[14] ampliamente utilizada y disponible, que permite extraer los sufijos y prefijos comunes de palabras literalmente diferentes pero con una raíz común que pueden ser consideradas como un sólo término. Tras este preprocesado se tiene un total de 21 730 palabras que constituyen el dominio de la aplicación.

Después de filtrados los documentos se utiliza el MEV como modelo de representación textual (descrito en la sección 2.3.1). Básicamente, cada documento se representa mediante un vector de términos. Cada término lleva asociado un coeficiente o peso que trata de expresar la importancia o grado de representatividad de ese término en ese vector o documento.

La red tendrá, un total de 21 730 unidades de entrada (correspondientes al total de términos utilizados) y 5 unidades de salida (correspondientes a las 5 categorías que se van a aprender).

3.2.3 Entrenamiento y evaluación de los resultados.

Una vez procesados los documentos y las categorías, se comienza el entrenamiento con el algoritmo LVQ.

Para realizar el entrenamiento y las pruebas se ha utilizado WEKA que es un paquete de software escrito en JAVA para implementar algoritmos de aprendizaje. A pesar de que resulta ser lento, posee grandes ventajas como las que se enuncian a continuación:

- De libre distribución.
- Multiplataforma.
- Tiene muchos algoritmos de regresión/clasificación.
- Incluye meta-algoritmos de aprendizaje (Bagging, AdaBoosting...)
- Tiene preprocesado de datos (selección, estadísticas,...)
- Incorpora herramientas para la visualización de los datos y resultados.
- Se distribuye también su código fuente JAVA.
- Se pueden añadir nuevas clases de clasificadores y filtros.
- Tiene versiones de consola y con interfaz gráfico.

[14] El algoritmo de Porter es uno de los algoritmos de extracción de raíces de palabras en inglés más utilizado. Disponible en Internet a través de la dirección: http://www.tartarus.org/~martin/PorterStemmer

Para realizar el entrenamiento con esta herramienta primeramente se han cargado los datos almacenados en un fichero de extensión .arff, que es el formato que soporta WEKA, el cual contiene la colección completa; de ella se toma el 75% para realizar el entrenamiento de la red y el 25% para comprobar la efectividad de esta. Durante el proceso la red irá tomando de manera aleatoria uno a uno los vectores del conjunto de entrenamiento. Para cada uno de ellos se selecciona como categoría ganadora aquella a la que pertenece el vector prototipo más cercano al vector de entrada. Luego se modifican los pesos según la ecuación 2.10 y se decrementa la *tasa de aprendizaje*. Se repite el proceso con el juego de patrones de aprendizaje hasta un número fijo de iteraciones o hasta que los pesos no varíen.

Para seleccionar la cantidad de neuronas de la capa oculta así como el valor de ajuste del parámetro $\alpha(t)$ se realizó un proceso de prueba y error; durante el cual se experimentó con diferentes valores de $\alpha(t)$ en el rango de 0,01 a 0,5. En la figura 3.2 se muestran los porcientos de aciertos[15] para diferentes valores de $\alpha(t)$ con una cantidad constante de 105 vectores prototipos donde se puede observar que para una *tasa de aprendizaje* de 0,1 se obtuvieron los mejores resultados.

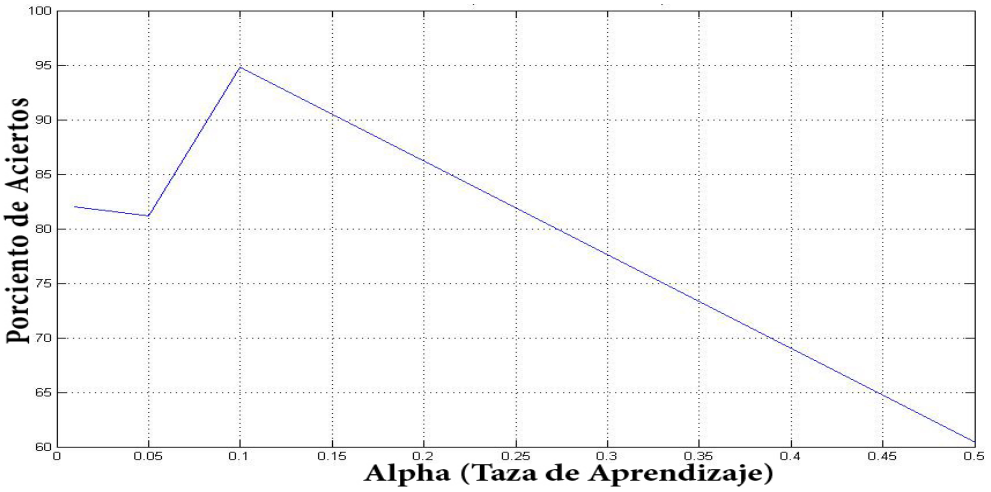

Figura 3. 2: Comportamiento de la red LVQ para diferentes tasas de aprendizaje.

Después de seleccionar la *tasa de aprendizaje* se realizaron otros experimentos, con una cantidad inicial de 10 vectores prototipos que se fueron incrementando en múltiplos de 10 hasta 160,

[15] En estos casos se habla de "porciento de aciertos" porque es uno de los resultados que da la herramienta WEKA que da una medida general bastante clara del comportamiento de la red.

tomándose para cada caso una cantidad de iteraciones 50 veces mayor que la cantidad de vectores anteriormente mencionados. Se lograron los mejores resultados con 100 y con 110 vectores prototipo, entonces se probó con un valor intermedio de 105. En la siguiente figura (3.3) se presenta el comportamiento de la red LVQ para el caso descrito y se demuestra que para una cantidad de 105 vectores prototipo se obtuvo el mejor resultado, con un 94,8% de aciertos y un 5,2% de instancias clasificadas incorrectamente.

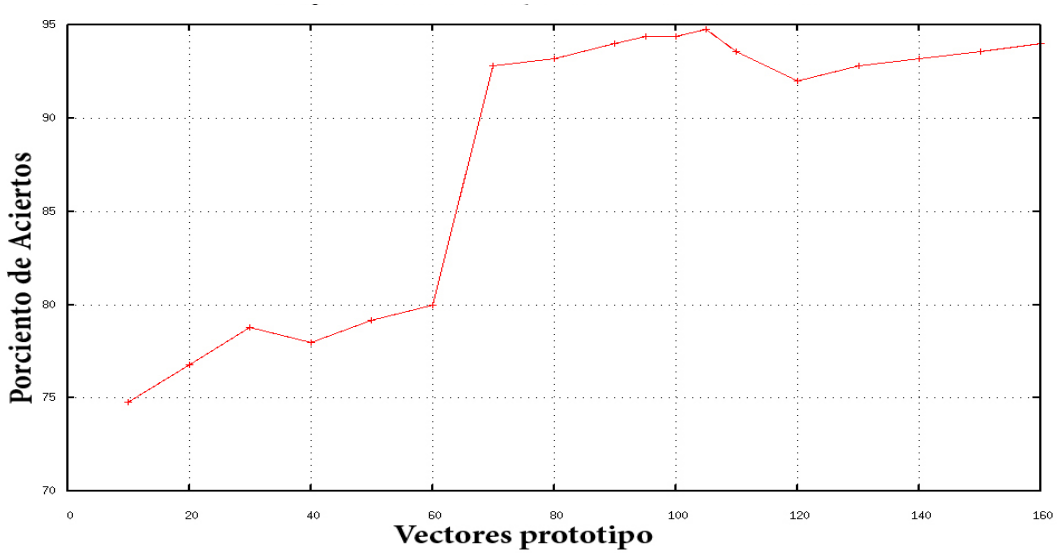

Figura 3. 3: Comportamiento de la red LVQ al aumentar la cantidad de vectores prototipos.

Para evaluar la efectividad del modelo propuesto se utilizan las métricas descritas en la sección 2.6. En los experimentos realizados con el algoritmo LVQ, para un valor de $\alpha(t)$ de 0,1 y 105 neuronas en la capa oculta, el mejor resultado se ha obtenido con la categoría “erotic”. La precisión para este caso es 0,98, el *recall* es 1 y la medida F es de 0,99. El peor caso se ha dado con la categoría “science_and_technology” con la que se tiene una precisión de 0,85, un *recall* de 0.92 y una medida F de 0.88. En la figura 3.4 se muestra lo descrito anteriormente.

Precision	Recall	F-Measure	Class
0.92	1	0.958	culture
1	0.923	0.96	economy
0.98	1	0.99	erotic
0.846	0.917	0.88	science_and_technology
1	0.907	0.951	sports

Figura 3. 4: Precisión, *Recall* y medida F para el mejor experimento.

La siguiente figura (3.5) muestra la precisión y el *recall* de la red LVQ para cada una de las categorías definidas al aumentar la cantidad de vectores prototipo.

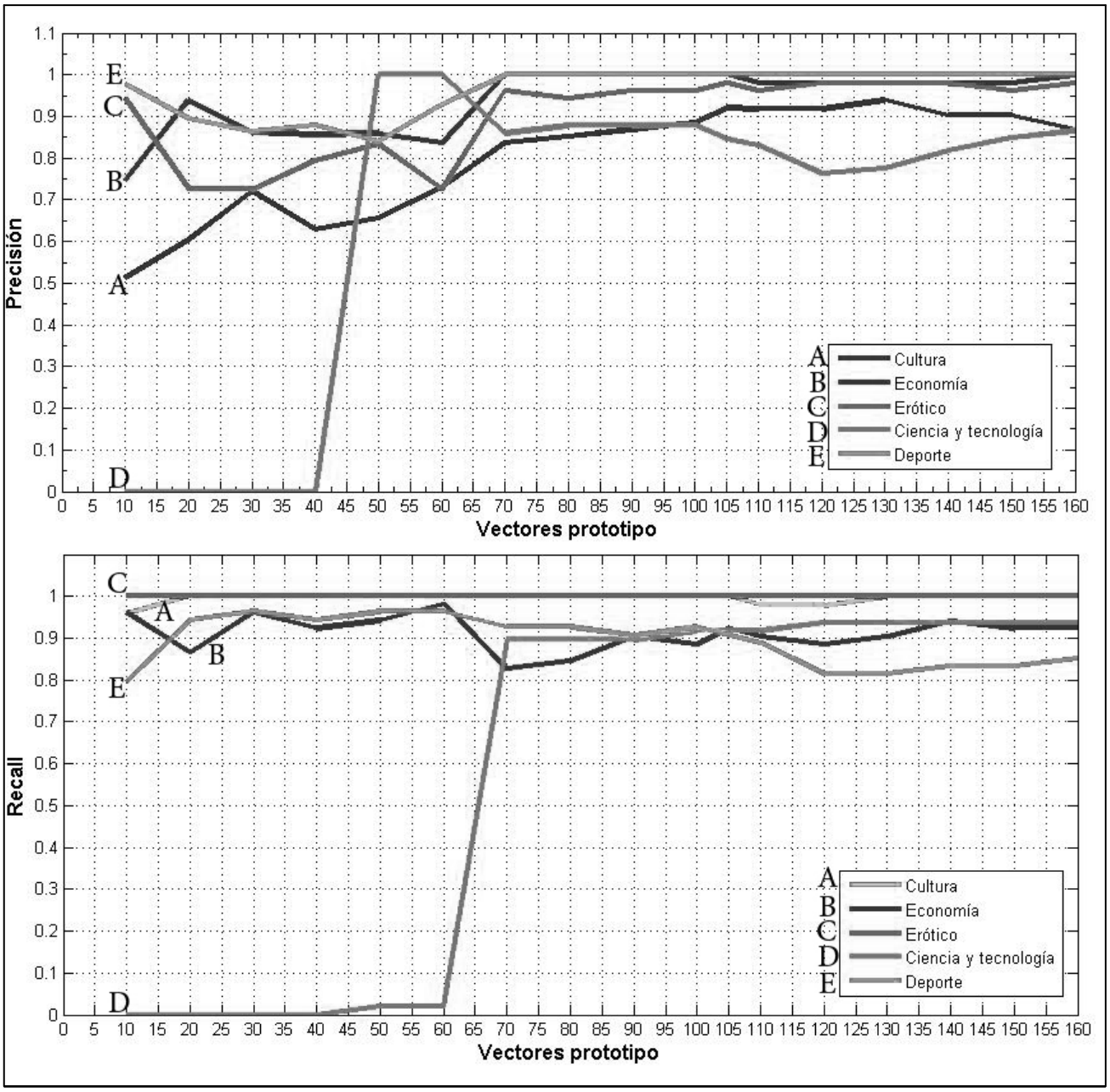

Figura 3. 5: Precisión y *Recall* de la red LVQ al aumentar la cantidad de vectores prototipo.

A continuación se muestra una gráfica que combina Precisión y *recall* para dar lugar a la medida F, que como se decía en la sección 2.6, permite comprobar con un único valor la efectividad del modelo (ecuación 2.16).

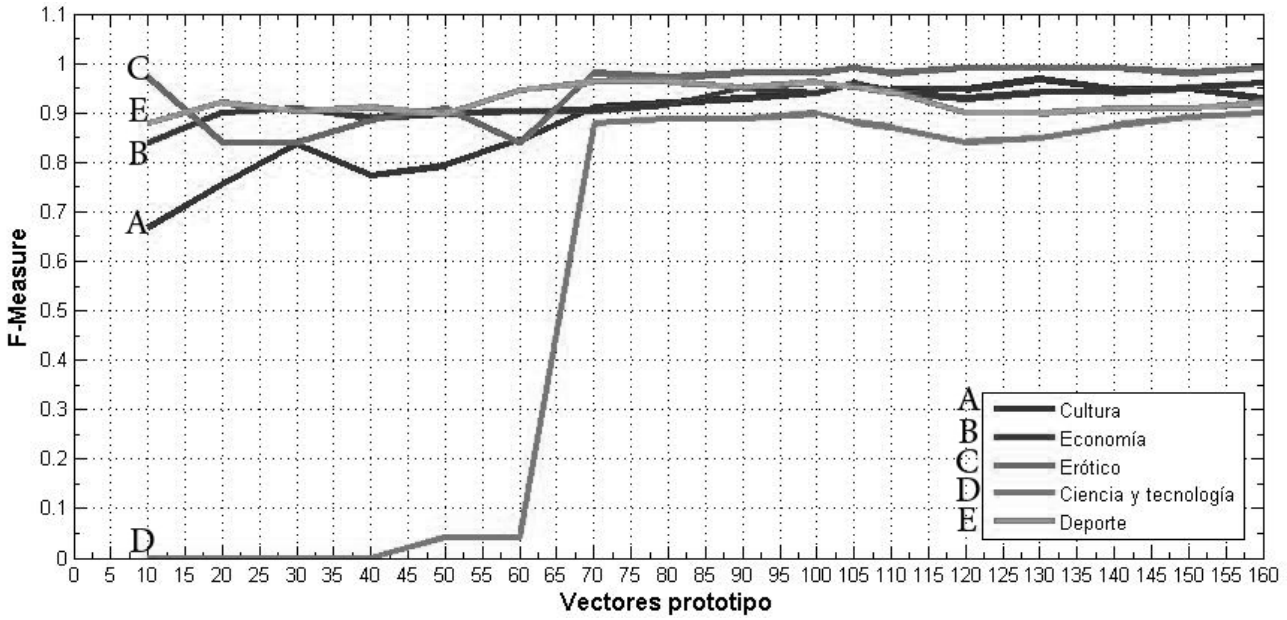

Figura 3. 6: Medida F de la red LVQ al aumentar la cantidad de vectores prototipo.

La efectividad media de un clasificador se puede medir de una manera global de varias formas; y una de estas es la precisión *macroaveraging* (sección 2.6) que consiste en calcular la media de la precisión de cada una de las categorías (ecuación 2.17). En la figura 3.7 se muestra la variación de este parámetro y se puede observar que el mejor caso se da con 105 vectores prototipo.

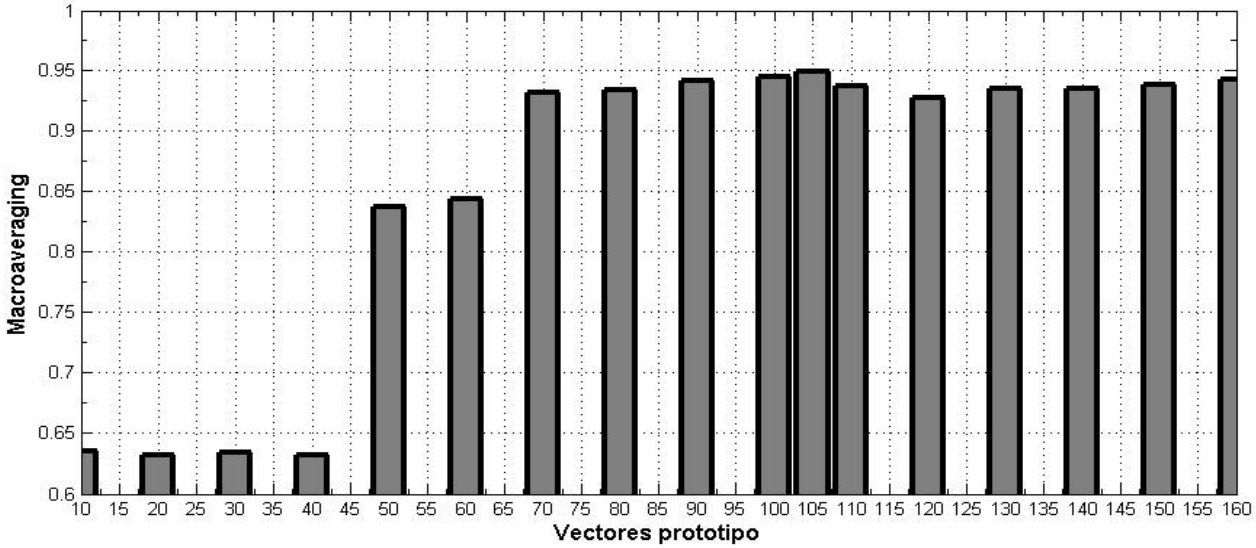

Figura 3. 7: Precisión *maroaveraging* de la red LVQ.

Para una mejor interpretación el WEKA permite visualizar los resultados obtenidos en diferentes variantes, incluyendo gráficas. Una de estas gráficas es la mostrada a continuación (figura 3.9), en la que se puede apreciar la representación de los errores de clasificación para el modelo descrito anteriormente.

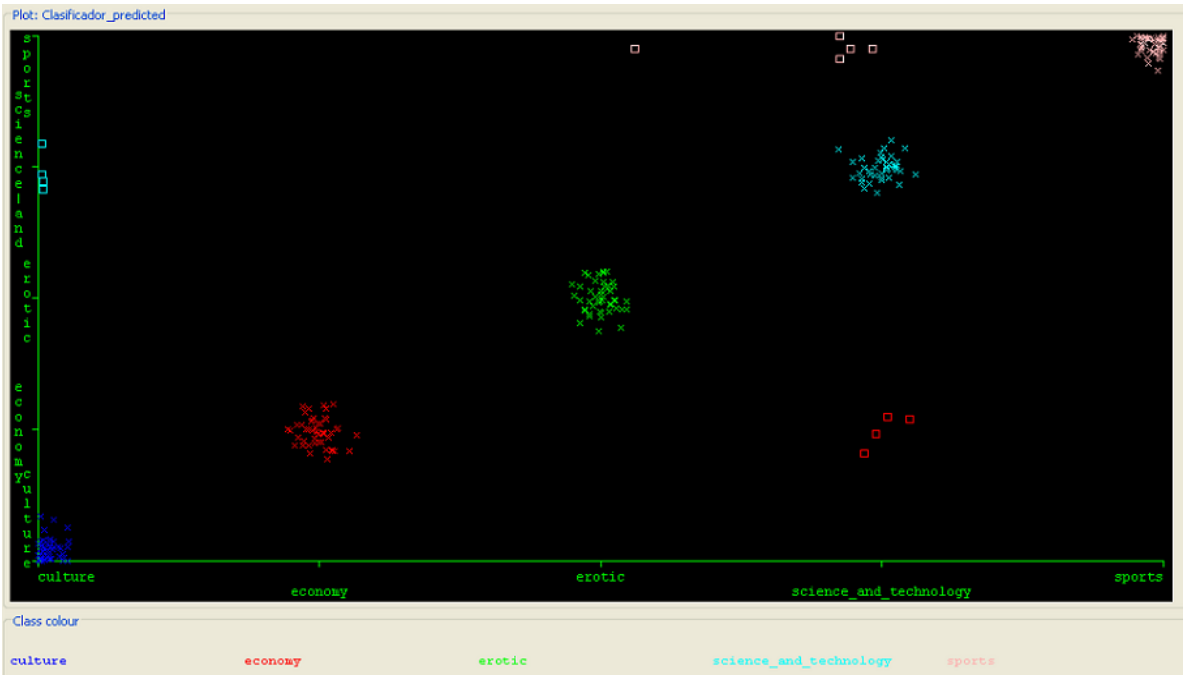

Figura 3. 8: Representación de los errores de clasificación.

Tanto por el eje X como por el eje Y se pueden observar las categorías, que además están representadas por los colores que se muestran en la parte inferior de la gráfica. Los cuadrados representan errores y las cruces aciertos. Como se explicó en el apartado 2.6, la precisión puede verse como la capacidad del clasificador para evitar clasificar en una categoría aquellos documentos que no deben ser categorizados con esa categoría (ecuación 2.11). Verticalmente entonces se puede ver el ejemplo de “cultura” que tiene una precisión de 0,92 (Figura 3.4), ya que existen documentos pertenecientes a “ciencia y tecnología” clasificados como “cultura”, más exactamente, cuatro documentos, según la matriz de confusión obtenida en el WEKA y que se muestra también en la figura 3.10. Del mismo modo se interpretan los resultados de *Precisión* para las otras categorías. Ahora bien, el *recall* que se explicaba también en el apartado 2.6 y que representa aquellos documentos que deberían ser clasificados en una determinada categoría, se puede ver en la gráfica horizontalmente, donde en el mismo ejemplo de la categoría “cultura”, se

obtiene un valor máximo (Figura 3.4) lo que equivale a que fueron categorizados con cultura los que debían ser clasificados como tal.

```
=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
 46  0  0  0  0 |  a = culture
  0 48  0  4  0 |  b = economy
  0  0 50  0  0 |  c = erotic
  4  0  0 44  0 |  d = science_and_technology
  0  0  1  4 49 |  e = sports
```

Figura 3. 9: Matriz de confusión.

La matriz de confusión da una información muy útil, porque no solo refleja los errores producidos sino también informa del tipo de estos. El número de columnas es el número de atributos y muestra la clasificación de las instancias; y las filas las categorías reales de los datos. Por lo que los elementos en la diagonal principal son los que ha acertado el clasificador y los demás son los errores.

Hasta aquí se ha podido demostrar que el algoritmo LVQ aplicado a la categorización de texto funciona eficientemente y que con el modelo obtenido después de los experimentos, se logran resultados realmente satisfactorios.

3.3. Conclusiones

En este capítulo se ha utilizado el algoritmo LVQ para categorizar una colección realizada manualmente de documentos extraídos de Internet. Se han realizado varios experimentos y mostrado gráficas del comportamiento de la red para diferentes parámetros. Para comprobar la efectividad del algoritmo se han utilizado las medidas clásicas de precisión, *recall*, precisión *macroaveraging* y la medida F. Finalmente se ha observado que el algoritmo LVQ funciona eficientemente y que se obtienen óptimos resultados.

Conclusiones Generales

Para lograr los objetivos de esta tesis se estudió el funcionamiento de las Redes Neuronales Artificiales aplicadas a la categorización de texto. También se creó una colección de entrenamiento de cinco categorías. Se utilizó el modelo de espacio vectorial como modelo de representación de información, se representaron los vectores prototipos asociados a cada categoría y se realizaron experimentos para entrenar y evaluar la red, mediante los cuales se determinaron los parámetros para los que esta alcanzó los mejores resultados. A partir del estudio realizado se obtuvo un modelo neuronal competitivo supervisado basado en el algoritmo LVQ que permite la categorización de texto para el proyecto FILPACON, cumpliéndose así con el objetivo general definido.

Recomendaciones

- Reducir el espacio de entrada a una dimensión menor.
- Estudiar en profundidad el aporte de los números a la colección de entrenamiento.
- Construir un clasificador con este modelo ya validado, ya que el resultado es excelente 95% de efectividad.
- Estudio de la categorización sensible al coste de las categorías.

Referencias

1) BARRO, M. J. *Computación Neuronal.* Universidad Santiago de Compostela, 1995.
2) BURGOS, F. J. P. *Herramientas en GNU/Linux para estudiantes universitarios. Redes Neuronales con GNU/Linux* Disponible en: http://softwarelibre.unsa.edu.ar/docs/descarga/2003/curso/htmls/redes_neuronales/index.html.
3) DARPA. Neural metwork study. *Defense Advanced Research Projects Agency (DARPA)*, 1988, nº
4) DAWSON, J. L. Suffix removal for word conflation. *Bulletin of the Association for Literary & Linguistic Computing*, 1974, vol. 2, nº 3, p. 33-46.
5) ESPINOZA, M. D. R. V. *Las Redes Neuronales Artificiales y su importancia como herramienta en la toma de decisiones.* Facultad de Ciencia Matemáticas. Universidad Nacional Mayor de San Marcos, 2002.
6) FERRÚS, I. M. J. R. N. A. *Unidad didactica "Viaje al universo neuronal".* Editado por: Tecnologia(Fecyt), F. E. P. L. C. Y. L. 2007, ISBN 978-84-690-4512-1.
7) GÁLVEZ, C. *El diccionario electrónico: un instrumento para la unificación de términos en la indización automática.* Universidad Alfonso X el Sabio ed. 2006, ISBN 1695-632X.
8) GOREN-BAR, T. K. D. L. D. Supervised Learning for Automatic Classification of Documents using Self-Organizing Maps. En *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries. ZŰrich, Switzerland. 2000.*
9) GPDS, G. D. P. D. D. S. *Redes Neuronales Artificiales* España: Disponible en: http://gpds.uv.es/nn/.
10) HAYKIN, S. *Neural Networks. A Comprehensive Foundation. .* Second Edition ed. Prentice Hall, 1999.
11) HAYKIN. *Neural networks: a comprehensive foundation* New York: Maxwell Macmillan, 1994.
12) KAFATI, E. G. *"Innovación bajo incertidumbre". Redes neuronales Ventajas y Desventajas* Disponible en: http://egkafati.bligoo.com/content/view/184582/Redes_neuronales_Ventajas_y_Desventajas.html.

13) KHIYAL, M. F. U. S. H. Classification of Textual Documents Using Learning Vector Quantization. *Information Technology*, 2007, vol. 6, nº 1, p. 154-159. ISSN 1812-5638.

14) KOHONEN, T. *Self-organization and associative memory.* Berlin: Springer- Verlag, 1995.

15) LEWIS, D. D. *Representation and learning in information retrieval.* Tesis Doctoral, Deparment of Computer and Information Science. Univ. of Massachusetts, Boston 1992.

16) LI, S. C. P. C. H. A Novel Algorithm for Text Categorization Using Improved Back-Propagation Neural Network En *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg, 2006, vol. 4223,

17) LIU, Y. Y. X. A re examination of text categorization methods. En *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval Berkeley, California, United States 1999*. p. 42 - 49

18) LOVINS, J. B. Development of a stemming algorithm. *Mechani-cal Translation and Computational Linguistics*, 1968, vol. 11, p. 22-31.

19) MARTÍN-VALDIVIA, M. G. V. L. A. U. L. M. T. Aprendizaje competitivo LVQ para la desambiguación léxica. *Procesamiento del Lenguaje Natural*, 2003, nº 31, p. 125-132 ISSN 1135-5948.

20) MERKL, A. R. D. Using self-organizing maps to organize document archives and to characterize subject matter: how to make a map tell the news of the world. En *10th International Conference Database and Expert Systems Applications(DEXA-99). Springer-Verlag, Berlin, Germany. 1999*. p. 302-311.

21) NIELSEN, R. H. *Neurocomputing.* Addison-Wesley, 1990. ISBN 9780201093551

22) PAICE, C. D. Another Stemmer. En *ACM SIGIR. 1990*. p. 56-61.

23) PÉREZ, J. M. *Introducción a la Neurocomputación* España: Universidad de Málaga, Disponible en: http://www.lcc.uma.es/~munozp/.

24) PORTER, M. F. An algorithm for suffix stripping. *Program*, 1980, vol. 14, p. 130-137.

25) SALTON, C. S. Y. A. W. G. Vector Space Model for Automatic Indexing. *Information Retrieval and Language Processing*, 1975, vol. 18 nº 11, ISSN 0001-0782

26) SANZ, J. C. C. P. J. M. G. H. M. D. B. R. E. P. Experimentos en indexación conceptual para la categorizacion de texto. En *Conferencia Ibero-Americana WWW/Internet 2004. Madrid, Spain,. October, 7-8 2004*. p. 251-258.

27) SAVIO, L. L. Y. D. *Learned text categorization by Backpropagation Neural Network.* Hong Kong University of Sciencie and Technology, 1996.

28) SEBASTIANI, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, march 2002 2002, vol. 34, nº 1, p. 1-47. ISSN 0360-0300

29) SKAPURA, J. A. F. D. M. *Neural Networks: Algorithms, Applications, and Programming Techniques* Addison-Wesley, 1991. ISBN 0201513765.

30) SRINIVASAN, M. E. R. P. Automatic Text Categorization Using Neural Networks. En *Advances in Classification Research, the 8th ASIS SIG/CR Classification Research Workshop. Medford:New Jersey. 1998*. p. 59-72.

31) TORRA-P, S. *Siniestralidad en seguros de consumo anual de las entidades de previsión social, La perspectiva probabilística y econométrica. Propuesta de un modelo econométrico neuronal para Cataluña.* Econometría, Estadística y Economía Española. Universidad de Barcelona, 2004.

32) UREÑA-LÓPEZ, M. G. V. M. T. M. V. L. A. Resolución de la ambigüedad mediante redes neuronales. *Procesamiento del Lenguaje Natural*, 2002, nº 29, p. 39-45. ISSN 1135-5948.

33) VALDIVIA. *Algoritmo LVQ aplicado a tareas de procesamiento de lenguaje natural.* Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga, 2004.

34) YANG, Y. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1999, vol. 1 nº 1-2, p. 69-90. ISSN 1386-4564.

35) ZULUAGA, H. S. I. M. I. A. C. A. *Turorial de Redes Neuronales* Universidad Tecnológica de Pereira (UTP). "Facultad de Ingeniería Eléctrica". Disponible en: http://ohm.utp.edu.co/neuronales/.

Bibliografía

1) ABHIMANYU LAD, Y. Y. Generalizing from Relevance Feedback using Named Entity Wildcards. En *Proceedings of the the sixteenth ACM conference on Information and Knowledge Management Lisbon, Portugal 2007* p. 721-730
2) BARRO, M. J. *Computación Neuronal.* Universidad Santiago de Compostela, 1995.
3) BRÍO, M. D. *Redes neuronales y sistemas difusos.* México: 2002. 399 p. ISBN 970-15-0733-9.
4) BURGOS, F. J. P. *Herramientas en GNU/Linux para estudiantes universitarios. Redes Neuronales con GNU/Linux* Disponible en: http://softwarelibre.unsa.edu.ar/docs/descarga/2003/curso/htmls/redes_neuronales/index.html.
5) CARLOS G. FIGUEROLA, J. L. A. B., ÁNGEL FRANCISCO ZAZO RODRÍGUEZ, EMILIO RODRÍGUEZ VÁZQUEZ DE ALDANA. *Herramientas para la investigación en Recuperación de Información: KARPANTA, un motor de búsqueda experimental* Universidad de Zaragoza ed. 2004, vol. 10 51-62 p. ISBN 1135-3716.
6) CHADE-MENG TAN, Y.-F. W., CHAN-DO LEE. The Use of Bigrams to Enhance Text Categorization. *Information Processing and Management*, July 2002, vol. 38, nº 4, p. 529-546. ISSN 0306-4573.
7) DARPA. Neural metwork study. *Defense Advanced Research Projects Agency (DARPA)*, 1988.
8) DAVID D. LEWIS, Y. Y., TONY G. ROSE, FAN LI RCV1: A New Benchmark Collection for Text Categorization Research. *Machine Learning Research*, 2004, vol. 5, p. 361 - 397 ISSN 1533-7928.
9) DAWSON, J. L. Suffix removal for word conflation. *Bulletin of the Association for Literary & Linguistic Computing*, 1974, vol. 2, nº 3, p. 33-46.
10) DTIL, U. L. D. T. E. I. D. L. L.-. *¿En qué lenguas habla Internet? Estadísticas 2007 sobre la presencia de lenguas latinas en la Red* Disponible en: http://dtil.unilat.org/LI/2007/es/resultados_es.htm.
11) ESPINOZA, M. D. R. V. *Las Redes Neuronales Artificiales y su importancia como herramienta en la toma de decisiones.* Facultad de Ciencia Matemáticas. Universidad Nacional Mayor de San Marcos, 2002.

12) F.SEBASTIANI. Automated text categorization: Tools, Techniques and Applications. En *April 2002.*

13) FERRÚS, I. M. J. R. N. A. *Unidad didáctica "Viaje al universo neuronal".* Editado por: Tecnologia(Fecyt), F. E. P. L. C. Y. L. 2007, ISBN 978-84-690-4512-1.

14) FRANCIS., W. N. Language corpora BC. En Svartvik, J. (editor). *Directions in Corpus Linguistics: proceedings of Nobel Symposium* 1992, p. 17-32.

15) GÁLVEZ, C. *El diccionario electrónico: un instrumento para la unificación de términos en la indización automática.* Universidad Alfonso X el Sabio ed. 2006, ISBN 1695-632X.

16) GOREN-BAR, T. K. D. L. D. Supervised Learning for Automatic Classification of Documents using Self-Organizing Maps. En *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries. ZŰrich, Switzerland. 2000*.

17) GPDS, G. D. P. D. D. S. *Redes Neuronales Artificiales* España: Disponible en: http://gpds.uv.es/nn/.

18) HAYKIN, S. *Neural Networks. A Comprehensive Foundation. .* Second Edition ed. Prentice Hall, 1999.

19) HAYKIN. *Neural networks: a comprehensive foundation* New York: Maxwell Macmillan, 1994.

20) JO, T. NTC (Neural Text Categorizer): Neural Network for Text Categorization. *IEEE Transactions on Neural Networks*, 2007, ISSN 1045-9227.

21) JOACHIMS, T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. En *The Fourteenth International Conference on Machine Learning 1997*. p. 143 – 151.

22) KAFATI, E. G. *"Innovación bajo incertidumbre". Redes neuronales Ventajas y Desventajas* Disponible en: http://egkafati.bligoo.com/content/view/184582/Redes_neuronales_Ventajas_y_Desventajas.html.

23) KHIYAL, M. F. U. S. H. Classification of Textual Documents Using Learning Vector Quantization. *Information Technology*, 2007, vol. 6, nº 1, p. 154-159. ISSN 1812-5638.

24) KOHONEN, T. *Self-organization and associative memory.* Berlin: Springer- Verlag, 1995.

25) ---. *Self-organizing Maps Third edition* 2001.

26) KRZYSZTOF J. CIOS, W. P. Data Mining: A Knowledge Discovery Approach. En 2007 p. pag 173.

27) KWAN YI, J. B. A Comparative Study on Feature Selection of Text Categorization for Hidden Markov Models 2004

28) L. ALFONSO URELÑA LÓPEZ, M. G. V., MANUEL DE BUENAGA RODRÍGUEZ, JOSÉ MARÍA GÓMEZ HIDALGO. *Resolución Automática de la Ambigüedad Léxica fundamentada en el Modelo del Espacio Vectorial usando Ventana Contextual Variable.* 1998,

29) LEWIS, D. D. *Representation and learning in information retrieval.* Tesis Doctoral, Deparment of Computer and Information Science. Univ. of Massachusetts, Boston 1992.

30) LI, S. C. P. C. H. A Novel Algorithm for Text Categorization Using Improved Back-Propagation Neural Network En *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg, 2006, vol. 4223,

31) LIU, Y. Y. X. A re examination of text categorization methods. En *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval Berkeley, California, United States 1999*. p. 42 – 49.

32) LOVINS, J. B. Development of a stemming algorithm. *Mechani-cal Translation and Computational Linguistics*, 1968, vol. 11, p. 22-31.

33) MANUEL GARCÍA VEGA, M. T. M. V., L. ALFONSO UREÑA LÓPEZ. Categorización de Textos Multilingües basada en Redes Neuronales. *Procesamiento del Lenguaje Natural*, 2001, p. 265-272. ISSN 1135-5948.

34) MARTÍN-VALDIVIA, M. G. V. L. A. U. L. M. T. Aprendizaje competitivo LVQ para la desambiguación léxica. *Procesamiento del Lenguaje Natural*, 2003, nº 31, p. 125-132 ISSN 1135-5948.

35) MERKL, A. R. D. Using self-organizing maps to organize document archives and to characterize subject matter: how to make a map tell the news of the world. En *10th International Conference Database and Expert Systems Applications (DEXA-99). Springer-Verlag, Berlin, Germany. 1999*. p. 302-311.

36) MIGUEL E. RUIZ, P. S. Hierarchical Text Categorization Using Neural Networks *Information Retrieval*, 2002, vol. 5, nº 1, p. 87-118.

37) MONICA ROGATI, Y. Y. High-Performing Feature Selection for Text Classification. En *Proceedings of the eleventh international conference on Information and knowledge management McLean, Virginia, USA. 2002*. p. 659 – 661.

38) NIELSEN, R. H. *Neurocomputing.* Addison-Wesley, 1990. ISBN 9780201093551.

39) PAICE, C. D. Another Stemmer. En *ACM SIGIR. 1990*. p. 56-61.

40) PENEDO, M. F. G. Departmento de Computación. Facultad de Informática. Universidad de A Coruña, Disponible en: http://carpanta.dc.fi.udc.es/~cipenedo/.

41) PÉREZ, J. M. *Introducción a la Neurocomputación* España: Universidad de Málaga, Disponible en: http://www.lcc.uma.es/~munozp/.

42) PORTER, M. F. An algorithm for suffix stripping. *Program*, 1980, vol. 14, p. 130-137.

43) SALTON, C. S. Y. A. W. G. Vector Space Model for Automatic Indexing. *Information Retrieval and Language Processing*, 1975, vol. 18 nº 11, ISSN 0001-0782.

44) SANZ, J. C. C. P. J. M. G. H. M. D. B. R. E. P. Experimentos en indexación conceptual para la categorizacion de texto. En *Conferencia Ibero-Americana WWW/Internet 2004. Madrid, Spain, October, 7-8 2004*. p. 251-258.

45) SAVIO L., Y. L., DIK LUN LEE. Feature reduction for Neural Network Based Text Categorization. En *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Systems for Advanced Applications. Hsinchu, Taiwan. 1999*. p. 195 – 202.

46) SAVIO, L. L. Y. D. *Learned text categorization by Backpropagation Neural Network.* Hong Kong University of Sciencie and Technology, 1996.

47) SEBASTIANI, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, march 2002 2002, vol. 34, nº 1, p. 1-47. ISSN 0360-0300.

48) SKAPURA, D. M. *Building Neural Networks.* 1996.

49) SKAPURA, J. A. F. D. M. *Neural Networks: Algorithms, Applications, and Programming Techniques* Addison-Wesley, 1991. ISBN 0201513765.

50) SRINIVASAN, M. E. R. P. Automatic Text Categorization Using Neural Networks. En *Advances in Classification Research, the 8th ASIS SIG/CR Classification Research Workshop. Medford:New Jersey. 1998*. p. 59-72.

51) SUSAN DUMAIS, J. P., MEHRAN SAHAMI, DAVID HECKERMAN. Inductive Learning Algorithms and Representations for Text Categorization En *Conference on Information and Knowledge Management. Bethesda, Maryland, United States. 1998*. p. 148-155.

52) TORRA-P, S. *Siniestralidad en seguros de consumo anual de las entidades de previsión social, La perspectiva probabilística y econométrica. Propuesta de un modelo econométrico neuronal para Cataluña.* Econometría, Estadística y Economía Española. Universidad de Barcelona, 2004.

53) UREÑA-LÓPEZ, M. G. V. M. T. M. V. L. A. Resolución de la ambigüedad mediante redes neuronales. *Procesamiento del Lenguaje Natural*, 2002, nº 29, p. 39-45. ISSN 1135-5948.

54) VALDIVIA. *Algoritmo LVQ aplicado a tareas de procesamiento de lenguaje natural.* Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga, 2004.

55) YANG, Y. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1999, vol. 1 nº 1-2, p. 69-90. ISSN 1386-4564.

56) ---. Using Corpus Statistics to Remove Redundant Words in the text Categorization. *Journal of the American Society for Information Science*, May 1996 1996, vol. 47 nº 5 p. pags. 357 - 369 ISSN 0002-8231.

57) YIMING YANG, J. O. P. A Comparative Study on Feature Selection in Text Categorization. En *Proceedings of the Fourteenth International Conference on Machine Learning 1997*. p. 412 – 420.

58) ZULUAGA, H. S. I. M. I. A. C. A. *Turorial de Redes Neuronales* Universidad Tecnológica de Pereira (UTP). "Facultad de Ingenieria Electrica". Disponible en: http://ohm.utp.edu.co/neuronales/.

Glosario

FILPACON: Sistema para regular el acceso a Internet de los usuarios de una red, diseñado para ser escalable según ámbito de instalación. El sistema permite la zonificación del Ciberespacio según categorías (pornografía, drogas, terrorismo, ciencia, deporte, salud, etc) y la gestión de cada usuario según políticas preestablecidas por un administrador.

Contenido Ilícito: Existe una completa serie de normas que limitan por distintas razones la utilización y la distribución de determinados contenidos. La infracción de dichas normas acarrea la ilicitud o ilegalidad de dichos contenidos.

Contenido Nocivo: Diversos tipos de materiales pueden constituir una ofensa a los valores o sentimientos de otras personas: contenidos que expresan opiniones políticas, creencias religiosas u opiniones sobre cuestiones raciales, etc. Lo que se considera nocivo depende de diferencias culturales. Cada país puede sacar sus propias conclusiones para la definición de la línea divisoria entre los permisible y lo que no lo es.

Contenido Adecuado: Aquellos que no entran en ninguna de las definiciones antes mencionadas. Aquellos que son perfectamente legales y que no afectan la moral de ninguna persona. En ambos casos, se parte de la base que existe una restricción esencial de carácter ético en su difusión por parte del responsable de su emisión.