

Universidad de las Ciencias Informáticas



Facultad 5

**Título: Edición del conjunto de datos de
entrenamiento de Redes Neuronales Artificiales**

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

Autor: Yosmany Siles Reina

Tutor: MsC. Marvyn Amado Márquez

Ing. Andy Trujillo Rivero

La Habana, de junio del 2011

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 _____.

Yosmany Siles Reina

Firma del Autor

Firma del Tutor

(Msc. Marvyn A. Márquez Rodríguez)

Firma del Co-Tutor

(Ing. Andy Trujillo Rivero)

DATOS DE CONTACTO

Tutor:

MsC. Marvyn Amado Márquez Rodríguez (mamarquez@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, Ciudad Habana.

Profesor Instructor Facultad 5 con 3 años de graduado.

4 años de experiencia en el tema de Redes Neuronales Artificiales.

Co-Tutor:

Ing. Andy Trujillo Rivero (arivero@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, Ciudad Habana.

Profesor Instructor Facultad 5 con 3 años de graduado.

2 años de experiencia en el tema de Redes Neuronales Artificiales.

La selección de los objetos de un dominio a incluir en un conjunto de entrenamiento es un problema presente en todos los modelos computacionales que realizan inferencias a partir de ejemplos. La edición del conjunto de entrenamiento se hace con el objetivo de eliminar los prototipos que inducen a una incorrecta clasificación supervisada, seleccionando un conjunto de referencia representativo y reducido. Las técnicas de edición, también producen la eliminación de prototipos, y con ello, la reducción de la matriz de aprendizaje.

En el presente trabajo se realiza un estudio relacionado con los métodos de edición existentes, las características de cada uno de ellos, así como la importancia de su utilización.

Palabras clave: aprendizaje, edición, inferencias.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
INTRODUCCIÓN	4
1.1 INTELIGENCIA ARTIFICIAL	4
1.1.2 Áreas de investigación.....	5
1.1.3 Áreas de aplicación	6
1.2 REDES NEURONALES ARTIFICIALES.	7
1.2.1 Ventajas de las Redes Neuronales Artificiales.	8
1.2.2 Aplicaciones de las RNA	9
1.2.3 Arquitectura de las RNA	10
1.2.4 Fases de operación de las RNA.....	11
1.2.4.1 Aprendizaje de las Redes Neuronales Artificiales	12
1.2.4.2 Fase de recuerdo	13
1.3 Framework de desarrollo	13
1.4 Lenguaje de programación.....	14
CONCLUSIONES DEL CAPÍTULO	15
CAPÍTULO 2: MÉTODOS DE EDICIÓN	17
INTRODUCCIÓN	17
2.1 REGLAS DE CLASIFICACIÓN.....	17
2.1.1 Regla del vecino más cercano.....	17
2.1.2 Regla de los k-vecinos más cercanos.....	18
2.1.3 Regla de los k-vecinos de centroide más próximo (clasificación envolvente).....	18
2.2 ALGORITMOS DE EDICIÓN	19
2.2.1 Algoritmo de edición de Wilson.	22
2.2.1.1 Edición de Wilson con distancia ponderada	23
2.2.2 Algoritmo de edición por Vecindad del centroide mas proximo.....	24
2.2.3 Algoritmo Backward Sequential Edition (BSE).....	25
2.2.4 Algoritmo Subconjunto Selectivo Modificado	26
2.2.5 Algoritmo de edición basado en Teoría de Conjuntos Aproximados.....	28
2.2.5.1 Consideraciones necesarias sobre la Teoría de Conjuntos Aproximados	28
2.2.5.2 Función de pertenencia aproximada.....	29
2.2.5.3 Algoritmo EditBRS	30
2.2.6 Algoritmo de edición basado en probabilidad de clases.....	30
2.2.7 Algoritmo de edición por partición.....	31
CONCLUSIONES DEL CAPÍTULO	32
CAPÍTULO 3: RESULTADOS EXPERIMENTALES	33
INTRODUCCIÓN	33
3.1 COMPARACIÓN TEÓRICA DE LOS MÉTODOS DE EDICIÓN	33
3.2 MODIFICACIONES A LOS MÉTODOS DE EDICIÓN.....	34
3.1 EDITOR DEL CONJUNTO DE ENTRENAMIENTO DE LAS RNA	35
3.3 RESULTADOS OBTENIDOS POR CADA UNO DE LOS MÉTODOS DE EDICIÓN MODIFICADOS EN CUANTO A REDUCCIÓN DE LA MUESTRA DE ENTRENAMIENTO	36
3.4 RESULTADOS OBTENIDOS POR CADA UNO DE LOS MÉTODOS DE EDICIÓN MODIFICADOS EN CUANTO A DISMINUCIÓN DEL ERROR DURANTE EL ENTRENAMIENTO.....	39
CONCLUSIONES DEL CAPÍTULO	44
CONCLUSIONES GENERALES	45
RECOMENDACIONES	46
BIBLIOGRAFÍA	47
GLOSARIO DE TÉRMINOS	48

INDICE DE FIGURAS

Figura 1 Regla de clasificación NN	17
Figura 2 Regla de clasificación k-NN	18
Figura 3 Regla de clasificación k-NCN	19
Figura 4 Estrategia incremental	20
Figura 5 Estrategia decremental.	21
Figura 6 Algoritmo de edición de Wilson	23
Figura 7 Algoritmo de edición del centroide más cercano	24
Figura 8 Algoritmo BSE.	26
Figura 9 Algoritmo subconjunto selectivo modificado	27
Figura 10 Fórmulas R-superior y R- inferior.	29
Figura 11 Función de pertenencia aproximada	29
Figura 12 Ecuación probabilidad de clases	31
Figura 13 Editor del conjunto de entrenamiento	35
Figura 14 Reducción del tamaño de la muestra de la base de datos Cáncer	37
Figura 15 Reducción del tamaño de la muestra de la base de datos Diabetes	37
Figura 16 Reducción del tamaño de la muestra de la base de datos Hebberman Survival.	38
Figura 17 Reducción del tamaño de la muestra de la base de datos mammographic mass.	38
Figura 18 Gráfica del error utilizando el algoritmo de edición de Wilson sobre el conjunto de entrenamiento cáncer	40
Figura 19 Gráfica del error utilizando el algoritmo de edición por Particiones sobre el conjunto de entrenamiento cáncer	40
Figura 20 Gráfica del error utilizando el algoritmo de edición por Probabilidad de clases sobre el conjunto de entrenamiento cáncer	41
Figura 21 Gráfica del error utilizando el algoritmo de edición de Wilson sobre el conjunto de entrenamiento mammographic mass.	42
Figura 22 Gráfica del error utilizando el algoritmo de edición por Particiones sobre el conjunto de entrenamiento mammographic mass	43
Figura 23 Gráfica del error utilizando el algoritmo de edición por Probabilidad de clases sobre el conjunto de entrenamiento mammographic mass	43

INDICE DE TABLAS

Tabla 1 Comparación teórica de los métodos de edición.	33
---	----

INTRODUCCIÓN.

Las tecnologías de la información y las comunicaciones (TIC) facilitan servicios, aplicaciones, herramientas y tecnologías. Dentro de ellas la industria del software alcanza una posición relevante, por las posibilidades de acelerar y facilitar el desarrollo de los países tanto en la economía, la educación, salud y otras esferas (Graells, 2008).

Cuba se ha propuesto no mantenerse aislada de los avances tecnológicos y alcanzar resultados sobresalientes en América Latina. En nuestro país, la industria del software es relativamente nueva, se han tomado determinadas iniciativas para intentar fortalecerla, ejemplo de ello es el surgimiento en el año 2002 del "Proyecto Futuro", el que luego pasó a ser la Universidad de las Ciencias Informáticas(UCI).

Las técnicas computacionales han marcado un nuevo paradigma en la creación. Sus aplicaciones, en la actualidad, van desde la industria de los juegos hasta las cadenas de producción de varias empresas. Para ello se han desarrollado múltiples técnicas como las relacionadas con la Inteligencia Artificial (IA). Una de las técnicas más conocidas es mediante las Redes Neuronales Artificiales (RNA) (Guzman, 2001).

Nuestra universidad no se ha quedado atrás en el uso de la IA y específicamente de las redes neuronales artificiales, basadas en las mismas se han realizado varias investigaciones, y se ha comenzado a hacer uso de ellas en varios proyectos, uno de los principales problemas que se producen durante el aprendizaje de dichas RNA es que aun no cuentan con métodos que sean capaces de eliminar del conjunto de entrenamiento los elementos atípicos que introducen ruido en la red que atentan contra la velocidad durante el aprendizaje y la capacidad de generalización de la red después de entrenada.

Otro de los problemas que poseen las RNA es la posibilidad real de la existencia de un número de patrones de entrenamiento que sea mayor a la cantidad que realmente necesita la red, entrando en un estado de sobreajuste o sobreentrenamiento de la red si es utilizado en su totalidad.

Teniendo en cuenta lo expuesto se define como problema de investigación científica: ¿Cómo eliminar del conjunto de entrenamiento los elementos considerados

como ruidosos, para reducir el tamaño de la muestra y disminuir el error durante el entrenamiento?

De esta manera el objeto de estudio se enmarca en el proceso de entrenamiento de las RNA.

Se precisa como objetivo general: editar el conjunto de datos de entrenamiento de las Redes Neuronales Artificiales. El campo de acción se enfoca en Métodos de edición sobre conjuntos de datos de entrenamiento para una RNA.

A partir del objetivo general, se derivan las tareas de la investigación que se enuncian a continuación:

1. Describir el proceso de entrenamiento de una RNA.
2. Identificar los problemas comunes en el proceso de entrenamiento.
3. Identificar y caracterizar métodos de edición descritos en la bibliografía para mejorar el proceso de entrenamiento.
4. Realizar una comparación entre el desempeño de los métodos de edición.
5. Comparar el desempeño de la RNA antes y después de utilizar el método de mejora del proceso de entrenamiento.

Por las razones antes expuestas se plantea como **idea a defender**.

Si se desarrolla con éxito la investigación se lograría editar el conjunto de entrenamiento de las RNA eliminando los prototipos que provocan problemas en el entrenamiento lo que nos permitiría acelerar el proceso de aprendizaje de las Redes Neuronales Artificiales.

El presente documento está estructurado de la siguiente manera: resumen, introducción, tres capítulos de contenido, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y anexos.

Capítulo 1: Fundamentación Teórica, se realiza un estudio donde se investiga acerca de los conceptos y características de la Inteligencia Artificial, Redes Neuronales Artificiales.

INTRODUCCIÓN

Capítulo 2: En este capítulo se realizará la descripción de los métodos de edición, además se caracterizan de forma detallada los mismos para mejorar el proceso de entrenamiento de las RNA.

Capítulo 3: Se realiza una comparación entre el funcionamiento de algunos de los métodos de edición, además se realiza una comparación entre una RNA entrenada con un conjunto sin editar y después de editar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se realiza un estudio acerca de los conceptos y características de la Inteligencia Artificial, Redes Neuronales Artificiales, así como sus ventajas y aplicaciones. Se hace referencia a los principales problemas que ocurren durante el proceso de aprendizaje de las Redes Neuronales Artificiales.

1.1 Inteligencia Artificial

Actualmente existen muchos conceptos que definen la Inteligencia artificial algunos de los cuales la enuncian como la rama de las ciencias de la computación que se ocupa de construir sistemas que permitan exhibir un comportamiento cada vez más inteligente (Artificial, 2006).

Otros hacen referencia a la IA como el arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia (Kurzweil, 1990). También puede ser definido como el estudio de cómo lograr que las computadoras realicen tareas que por el momento, los humanos hacen mejor (Elaine Rich, 1991). Son muchos los conceptos que se le dan a esta ciencia pero todos convergen en que se refiere al estudio de cómo darle la inteligencia del hombre a la máquina.

Actualmente la Inteligencia Artificial es una rama que tiene una alta aplicación en diferentes ámbitos, debido a que incluye diferentes campos de desarrollo entre los que destacan la robótica, usado mayormente en el campo de la industria, la comprensión de lenguajes y traducción, reconocimiento de palabras y aprendizaje de máquinas, sistemas computacionales expertos, que reproducen el comportamiento humano de manera eficiente.

La misma posee varias características entre las que destacan (Tabar, 2005):

- Uso de caracteres no numéricos para especificar los diferentes algoritmos. Las diferentes sentencias son evaluadas mediante álgebra y lógica.
- El comportamiento de los programas no está explícitamente definido por los algoritmos sino que el flujo de ejecución depende del problema en sí mismo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- El razonamiento está basado en el conocimiento, los programas de Inteligencia Artificial pueden distinguir entre el programa de razonamiento o motor de inferencia y base de conocimientos dándole la capacidad de explicar discrepancias entre ellas.

Cabe destacar que en cuanto al funcionamiento de la IA existen 2 teorías:

- Construir réplicas de la compleja red neuronal del cerebro humano.
- Intentar imitar el comportamiento del cerebro humano con un computador.

1.1.2 Áreas de investigación.

Actualmente son muchas las áreas de la investigación relacionadas con la Inteligencia Artificial entre ellas podemos encontrar (Fernández, 2004):

- La representación del conocimiento, que busca en el descubrimiento de métodos expresivos y eficientes describir información sobre aspectos del mundo real.
- Los métodos de aprendizaje automático, que extienden las técnicas estadísticas con el fin de posibilitar la identificación de un amplio rango de tendencias generales a partir de un conjunto de datos de entrenamiento.
- El campo de la planificación, que enfrenta el desarrollo de algoritmos que construyen y ejecutan automáticamente secuencias de comandos primitivos con el fin de alcanzar ciertas metas de alto nivel.
- Los trabajos en el área de razonamiento posible, que hacen uso de principios estadísticos para desarrollar codificaciones de información incierta.
- El estudio de las arquitecturas de agentes, que busca la integración de otras áreas de la IA con el objeto de crear agentes inteligentes, entidades robustas capaces de comportamiento autónomo y en tiempo real.
- La coordinación y colaboración multiagentes, que ha permitido el desarrollo de técnicas para la representación de las capacidades de otros agentes y la especificación del conocimiento necesario para la colaboración entre ellos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- El desarrollo de ontologías, que persigue la creación de catálogos de conocimiento explícito, formal y multipropósito, que puedan ser utilizados por sistemas inteligentes.
- Los campos de procesamiento de voz y lenguaje, que buscan la creación de sistemas que se comunican con la gente en su lenguaje.
- La síntesis y comprensión de imágenes, que conduce a la producción de algoritmos para el análisis de fotografías, diagramas y videos, así como también de técnicas para el despliegue visual de información cuantitativa y estructurada.

1.1.3 Áreas de aplicación

Son muchas las áreas donde tienen aplicación la Inteligencia Artificial dentro de las mismas tenemos (Fernández, 2004):

- Gestión y control: análisis inteligente, fijación de objetivos.
- Fabricación: diseño, planificación, programación, monitorización, control, gestión de proyectos, robótica simplificada y visión computarizada.
- Educación: adiestramiento práctico, exámenes y diagnóstico.
- Ingeniería: diseño, control y análisis
- Equipamiento: diseño, diagnóstico, adiestramiento, mantenimiento, configuración, monitorización y ventas.
- Cartografía: interpretación de fotografías, diseño, resolución de problemas cartográficos.
- Profesiones: abogacía, medicina, contabilidad, geología, química.
- Software: enseñanza, especificación, diseño, verificación, mantenimiento.
- Sistemas de armamento: guerra electrónica, identificación de objetivos, control adaptativo, proceso de imágenes, proceso de señales.
- Proceso de datos: educación, interface en lenguaje natural, acceso inteligente a datos y gestores de bases de datos, análisis inteligente de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Finanzas: planificación, análisis, consultoría.

1.2 Redes Neuronales Artificiales.

Las Redes Neuronales Artificiales son programas de la IA capaces de simular algunas de las funciones de aprendizaje del ser humano. Una red neuronal obtiene experiencia analizando automática y sistemáticamente los datos para determinar reglas de comportamiento; con base en ellas, puede realizar predicciones sobre nuevos casos (Fernández, 2004).

A continuación se muestran algunos conceptos de RNA dados por varios autores:

- Se define a las RNA como un esquema de computación distribuida inspirada en la estructura del sistema nervioso de los seres humanos. Las mismas están formadas por una gran cantidad de neuronas, estas no suelen denominarse neuronas artificiales sino nodos o unidades de salida. Un nodo o neurona cuenta con una cantidad variable de entradas que provienen del exterior. A su vez dispone de una sola salida que transmitirá la información al exterior o hacia otras neuronas (Salas, 2007).
- Modelo computacional paralelo compuesto por unidades procesadoras adaptativas con una alta interconexión entre ellas (Soria, 2001).
- Un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas (Hetch, 1988).
- Sistema de procesado de la información que hacen uso de algunos de los principios que organizan la estructura del cerebro humano (Soria, 2001).
- Sistema de procesado de la información que tiene características de funcionamiento comunes con las Redes Neuronales biológicas (Soria, 2001).
- Es un sistema de procesamiento de información compuesto por un gran número de elementos de procesamiento (neuronas), profusamente conectados entre sí a través de canales de comunicación (Reguera, 1995).

Las conexiones a las que se refieren en el último concepto, establecen una estructura jerárquica y permiten la interacción con los objetos del mundo real tratando de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

emular al sistema nervioso biológico. A diferencia de la computación tradicional, basada en algoritmos predecibles, la computación neuronal permite desarrollar sistemas que resuelvan problemas complejos cuya formalización matemática es sumamente difícil.

Podemos llegar a la conclusión a través de los conceptos mencionados anteriormente que las Redes Neuronales Artificiales son sistemas que poseen características similares al cerebro humano y tratan de imitar su comportamiento.

1.2.1 Ventajas de las Redes Neuronales Artificiales.

Debido a su constitución y a sus fundamentos las Redes Neuronales Artificiales presentan un gran número de características semejantes a las del cerebro, por ejemplo son capaces de aprender de la experiencia, es decir generalizar de casos conocidos a nuevos casos. Esto permite que ofrezca múltiples ventajas tales como (Fuente, 2009):

- Aprendizaje adaptativo: esta es quizás la característica más importante de las RNA, ya que pueden comportarse en función de un entrenamiento con una serie de ejemplos ilustrativos. De esta forma, no es necesario elaborar un modelo a priori, ni establecer funciones probabilísticas. Una RNA es adaptativa porque puede modificarse constantemente con el fin de adaptarse a nuevas condiciones de trabajo.
- Auto organización: Una Red Neuronal Artificial puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos: En la computación tradicional la pérdida de un fragmento pequeño de información puede acarrear comúnmente la inutilización del sistema. Las RNA poseen una alta capacidad de tolerancia a fallos. La tolerancia a fallos se entiende aquí en dos sentidos: primero, las redes pueden reconocer patrones de información con ruido, distorsión o incompletos (tolerancia de fallos respecto de los datos); y segundo, pueden seguir trabajando (con cierta degradación) aunque se destruya parte de la red (tolerancia a fallos respecto de la estructura). La explicación de este fenómeno se encuentra en que, mientras la computación tradicional almacena la información en espacios únicos, localizados y direccionales, las redes neuronales lo hacen de forma distribuida y con un alto grado de redundancia.
- Operación en tiempo real: de todos los métodos existentes, la RNA son las más indicadas para el reconocimiento de patrones en tiempo real, debido a que trabajan

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

en paralelo actualizando todas sus instancias simultáneamente. Es importante destacar que esta característica solo se aprecia cuando se implementan redes con hardware especialmente diseñado para el procesamiento en paralelo.

- Fácil inserción en la tecnología existente: es relativamente sencillo obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilita la integración modular en los sistemas existentes.
- Son sistemas distribuidos no lineales: Una neurona es un elemento no lineal por lo que una interconexión de ellas también será un dispositivo no lineal. Esta propiedad permitirá la simulación de sistemas no lineales y caóticos, que con los sistemas clásicos lineales, no se puede realizar.
- Adaptabilidad: Una Red Neuronal Artificial tiene la capacidad de modificar los parámetros de los que depende el funcionamiento de acuerdo con los cambios que se produzcan en su entorno de trabajo.
- Pueden establecer relaciones no lineales entre datos: Las RNA son capaces de relacionar conjuntos de datos mediante relaciones complejas.

1.2.2 Aplicaciones de las RNA

Existe una gran variedad de RNA, cada una de ellas tiene una aplicación particular más apropiada. Algunas de las aplicaciones se presentan a continuación (Slideshare, 2008):

- Biología: Clasificación de seres vivos y fósiles en sus leyes genéticas, anatómicas, fisiológicas y taxonómicas.
- Empresa: Evaluar probabilidades de formaciones geológicas y petrolíferas, optimización de plazas y horarios en líneas de vuelo, identificación de candidatos para ubicaciones específicas.
- Medio Ambiente: Analizar tendencias y patrones, previsión del tiempo.
- Finanzas: Prever la evolución de los precios, valoración de riesgo de los créditos, identificación de falsificaciones.
- Manufactura: Creación de robots automatizados y sistemas de control.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Medicina: Analizadores del habla para ayudar en la audición de sordos profundos, predicción de reacciones adversas en los medicamentos, diagnóstico y tratamiento a partir de síntomas a partir de síntomas y datos analíticos.
- Militares: Clasificación de las señales de radar, creación de armas inteligentes, reconocimiento y seguimiento de tiro al blanco.

1.2.3 Arquitectura de las RNA

La arquitectura de la red neuronal se forma mediante la conexión múltiple de varios procesadores, siendo este un sistema adaptativo que posee un algoritmo para ajustar los pesos para lograr el correcto desempeño basado en las muestras de entrenamiento (Salas, 2007) .Se definen dos modelos de arquitecturas:

- Redes Monocapa: Se establecen conexiones laterales, cruzadas o auto recurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Esto se debe a que los valores de entrada no pertenecen a la red. Las mismas se utilizan generalmente en la auto asociación (regenerar información de entrada que se presenta a la red de forma incompleta o distorsionada).
- Redes Multicapa: Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior que esté más cerca de la entrada de la red y envían señales de salida a una capa posterior que esté más cerca a la salida de la red. A estas conexiones se le denomina conexiones hacia adelante o feedforward.

Por otro lado, existe la posibilidad de conectar la salida de las neuronas de las capas posteriores a la entrada de las capas anteriores, a estas conexiones se las denomina conexiones hacia atrás o feedback.

Estas posibilidades permiten que exista una clasificación diferente entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante o feedforward, y las redes con conexiones tanto hacia adelante como hacia atrás o feedforward/feedback. Lo que da lugar a estructuras neuronales que pueden ser clasificadas por su tipo de conexión o su grado de conexión (Soria, 2001).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Redes neuronales Artificiales no recurrentes: También conocidas como redes en cascada, la propagación de las señales se realiza unidireccionalmente de una capa a otra, la información fluye desde los datos de entrada a las capas ocultas y de estas a las de salida (Nores, 2005).
- Redes neuronales Artificiales recurrentes: La misma está caracterizada por la existencia de lazos de retroalimentación. Dichos lazos pueden ser entre neuronas de diferentes capas, neuronas de la misma capa, o entre una misma neurona.
- Redes neuronales Artificiales totalmente conectadas: En este caso todas las neuronas de una capa se encuentran conectadas con las de la capa siguiente.
- Redes neuronales Artificiales conectadas: En este caso no se da la conexión total entre las neuronas de diferentes capas.

Podemos llegar a la conclusión de que una red neuronal artificial es un sistema computacional que está caracterizado por (Rojas, 2009):

- Un conjunto de unidades elementales, cada una de las cuales posee bajas capacidades de procesamiento
- Un alto grado de paralelismo.
- Una densa estructura interconectada usando enlaces ponderados
- Parámetros libres que deben ser ajustados para satisfacer los requerimientos de desempeño.

Se hace necesario puntualizar que la propiedad más importante de las redes neuronales artificiales es su capacidad de aprendizaje a partir de las muestras de entrenamiento.

1.2.4 Fases de operación de las RNA

Las RNA pueden trabajar en dos modos: modo de aprendizaje o entrenamiento y modo de recuerdo o ejecución.

1.2.4.1 Aprendizaje de las Redes Neuronales Artificiales

El aprendizaje es el proceso mediante el cual los parámetros libres de una red neuronal son ajustados a través de un proceso continuo de estimulación del entorno en donde se sitúa el sistema (González, 2003).

Existen dos tipos de reglas de entrenamiento de las redes neuronales artificiales: las de aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado lo que da a lugar a una de las clasificaciones que se realizan de las RNA.

Redes con Aprendizaje Supervisado: El aprendizaje se efectúa a través de un entrenamiento controlado por un agente externo que determina la respuesta que debe brindar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida se aproxime a la deseada (González, 2003).

Existen tres formas de llevar a cabo este tipo de aprendizaje:

- Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error.
- Aprendizaje por refuerzo: Se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. La función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada.
- Aprendizaje estocástico: Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

Redes con Aprendizaje No Supervisado: Las redes no necesitan la influencia externa para ajustar los pesos de las conexiones entre neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada es o no correcta, así que existen varias posibilidades en cuanto a la interpretación de la salida de estas redes (González, 2003).

En general en este tipo de aprendizaje se suelen considerar dos tipos:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Aprendizaje Hebbiano: Consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación.
- Aprendizaje competitivo y cooperativo: Las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada.

La principal diferencia entre ambos es la existencia o no de un agente externo en el proceso de aprendizaje de la red.

1.2.4.2 Fase de recuerdo

Generalmente (aunque no en todos los modelos), una vez que el sistema ha sido entrenado, el aprendizaje “se desconecta”, por lo que los pesos y la estructura quedan fijos, estando la red neuronal ya dispuesta para procesar datos. Este modo de operación también es denominado “recall”.

1.3 Framework de desarrollo

Qt es un framework de desarrollo para aplicaciones multiplataforma para el desarrollo de aplicaciones en C++, también puede ser utilizado en otros lenguajes. Es una biblioteca para el desarrollo de interfaces gráficas y también para el desarrollo de programas sin interfaz gráfica como herramientas de consola y servidores (Qt, 2010).

Este framework está compuesto por varias librerías y permite la creación de formularios visualmente con QtDesigner, además garantiza un acceso rápido a la documentación mediante el QtAssistant y una traducción rápida mediante el QtLinguist.

La Interfaz de Programación de Aplicaciones(API) de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML; cuenta con una API multiplataforma unificada para la manipulación de archivos y otras para el manejo de ficheros, además de estructuras de datos tradicionales.

Algunas de las características se mencionan a continuación (QtCreator, 2010):

- Se distribuye bajo una licencia libre GPL (o QPL), además de la LGPL, que nos permite incorporar las QT en nuestras aplicaciones open-source.
- Se encuentra disponible para una gran número de plataformas: Linux, MacOs X, Solaris, HP-UX, UNIX con X11, Windows.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Es orientado a objetos, lo que facilita el desarrollo de software. El lenguaje para el que se encuentra disponible es C++ aunque han aparecido versiones (conocidas como bindings) para C, Python (PyQt), Java (Qt Jambi), Perl (PerlQt), entre otros.
- Es una biblioteca que se basa en los conceptos de widgets (objetos), Señales-Slots y Eventos (Ej.: clic del ratón).
- Las señales y los slots es el mecanismo para que unos widgets se comuniquen con otros.
- Los widgets pueden contener cualquier número de hijos. El widget superior (también conocido como top-level), puede ser cualquiera, sea ventana, botón, etc.
- Compatibilidad multiplataforma con un sólo código fuente.
- Fácil de internacionalizar.
- Arquitectura lista para plugins.

Teniendo en cuenta las características antes mencionadas llegamos a la conclusión de Qt posee tres grandes ventajas (Document, 2010):

- Qt es completamente gratuito para aplicaciones de código abierto, y una de las licencias mencionadas anteriormente, la LGPL, permite que se utilice gratuitamente con fines comerciales.
- Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X, Solaris, etc.) como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa.
- Tiene una extensa librería con clases y herramientas para la creación de aplicaciones con interfaz de usuario enriquecidas. Estas librerías y clases están bien documentadas, son muy fáciles de usar y tienen una gran herencia de programación orientada a objetos lo cual hace de la programación de interfaces gráficas un trabajo más ameno.

1.4 Lenguaje de programación

Como lenguaje de programación se utiliza el C++, debido que con el mismo se puede llevar a cabo el desarrollo de cualquier tipo de programa. Además C++ brinda muchas

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

facilidades para la programación orientada a objetos y para el uso de plantillas o programación genérica. Está considerado como un potente lenguaje, ya que trabaja tanto a alto como a bajo nivel por lo que el código resultante de su compilación es muy eficiente. Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel como son la identificación de tipos en tiempo de ejecución (RTTI) y la posibilidad de redefinir operadores.

La posibilidad de orientar la programación a objetos que nos ofrece este lenguaje permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además permite la reutilización del código de una manera más lógica y productiva. También es de alta portabilidad y brevedad.

Otra importante característica de C++ es la programación modular ya que una entidad de aplicación en C++ puede estar hecha con varios ficheros de código fuente que son compilados por separado y después unidos. Esta característica permite unir código en C++ con código producido en otros lenguajes.

Ventajas (C++, 2011):

Las principales ventajas que presenta el lenguaje C++ son:

- **Difusión:** al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, páginas web, dedicados a él.
- **Versatilidad:** es un lenguaje de propósito general, por lo que se puede emplear para cualquier tipo de problema.
- **Portabilidad:** el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- **Eficiencia;** es uno de los lenguajes más rápidos en cuanto a ejecución.
- **Herramientas:** existen una gran cantidad de compiladores, depuradores librerías, entre otros.

CONCLUSIONES DEL CAPÍTULO

En el capítulo se definieron los conceptos, ventajas y aplicaciones de la Inteligencia Artificial y Redes Neuronales Artificiales (RNA) éstas tienen gran aplicación en ramas como la Biología y la Medicina sirviendo de apoyo al desarrollo científico. Se trataron varios

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

aspectos relacionados con el aprendizaje de las RNA un ejemplo de ello son los principales problemas que ocurren durante el entrenamiento de una RNA que influyen negativamente en la capacidad de generalización de la red entrenada.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Introducción

En este capítulo se realizará la descripción de los métodos de edición, además se caracterizan de forma detallada los mismos para mejorar el proceso de entrenamiento de las RNA.

2.1 Reglas de clasificación.

A continuación se exponen una serie de reglas de clasificación que son utilizadas en los algoritmos de edición del conjunto de datos de entrenamiento que serán estudiados posteriormente.

2.1.1 Regla del vecino más cercano

Una de las reglas de clasificación más difundida en la literatura es la del vecino más cercano (NN), la idea fundamental sobre la cual se centra la misma es que las muestras de una misma clase se encontrarán probablemente próximas en el espacio (Vázquez, 2008).

Dados el conjunto $\{X, \Theta\} = \{(x_1, \theta_1), (x_2, \theta_2), (x_3, \theta_3), \dots, (x_n, \theta_n)\}$, que contiene N prototipos, y una nueva muestra x , de la que se desconoce su clase.

Sea $(x', \theta') \in \{X, \Theta\}$, el prototipo más próximo a la muestra x , entonces la regla NN se podrá escribir valiéndonos de la expresión (Vázquez, 2008):

$$\delta_{NN}(x) = \theta' \Leftrightarrow d(x, x') = \min_{i=1, 2, \dots, N} d(x, x_i)$$

Figura 1 Regla de clasificación NN

Cabe destacar que el análisis asintótico de la regla NN, permite afirmar que, al menos, la mitad de la información sobre la pertenencia de un objeto a una cierta clase se encuentra en su vecino más próximo. Resultado este que no depende de la métrica que se haya utilizado para la determinación del vecino más cercano (Vázquez, 2008).

CAPÍTULO 2: MÉTODOS DE EDICIÓN

2.1.2 Regla de los k-vecinos más cercanos.

Es lógico pensar que con la aplicación de la regla NN, no se está aprovechando al máximo toda la información que se podría obtener del conjunto de entrenamiento debido a que esta regla solamente utiliza el primer vecino más próximo.

Con el objetivo de suplir dicha aparente dificultad surge una mejora a la regla de clasificación NN, la cual consiste en una modificación de la forma empleada para determinar la clase a la que pertenece determinada muestra, utilizando no solo su primer vecino más cercano, sino también un cierto número de prototipos (k) que se encuentren en un entorno próximo a la muestra a clasificar (Vázquez, 2008).

Teniendo en cuenta la idea planteada anteriormente a partir de cierto conjunto de entrenamiento $\{X, \Theta\} = \{(x_1, \theta_1), (x_2, \theta_2), (x_3, \theta_3), \dots, (x_n, \theta_n)\}$, podremos definir la regla de clasificación de los k-vecinos más cercanos (k-NN) de la siguiente forma (Vázquez, 2008):

$$\delta_{k\text{-NN}}(x) = w_i \Leftrightarrow d(x, P_i) = \min_{j=1,2,\dots,M} d_k(x, P_j)$$

Figura 2 Regla de clasificación k-NN

Podemos señalar que el significado de la expresión anterior no es más, que la clase asignada a la muestra x será la clase más votada entre los k-vecinos más cercanos del conjunto de entrenamiento.

Generalmente, en problemas prácticos donde se aplica esta regla de clasificación se debe tomar un número k impar de vecinos para evitar posibles empates.

2.1.3 Regla de los k-vecinos de centroide más próximo (clasificación envolvente).

La vecindad de centroide más próximo a un punto puede ser determinada siguiendo un método sencillo, en el que el primer vecino de un punto p se corresponde con su vecino más próximo, mientras que los sucesivos vecinos se tomarán de manera que minimicen la distancia entre p y el centroide de todos los vecinos seleccionados hasta el momento (Vázquez, 2008).

Así, si calculamos el k-ésimo vecino a partir de los k-1 vecinos previamente elegidos por el principio de centroide más próximo, conseguiremos cumplir con los criterios de distancia y simetría.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

En realidad, la condición de distancia se satisface por el hecho de tomar el vecino más próximo como el punto de partida para el cálculo de los posteriores $k-1$ vecinos.

Cabe destacar que, como consecuencia del criterio de centroide que se está utilizando, todos los vecinos k -NCN seleccionados se situaran alrededor del punto p , es decir, de alguna forma se consigue que dicho punto quede rodeado por sus k vecinos.

Supongamos que contamos con un conjunto de de datos formado por N prototipos formados por M clases diferentes $\{X, \Theta\} = \{(x_1, \theta_1), (x_2, \theta_2), (x_3, \theta_3), \dots (x_n, \theta_n)\}$, y sea (x', θ') el vecino más cercano de una muestra $x \in X$, entonces la regla de clasificación k -NCN quedaría de la siguiente forma (Vázquez, 2008):

$$\delta_{k\text{-NCN}}(x) = \omega_i \Leftrightarrow d_k(x, P_i) = \min_{i=1, \dots, M} d_k(x, P_i)$$

Figura 3 Regla de clasificación k -NCN

El significado de la expresión anterior consistirá en que la clase asignada a la muestra x corresponderá a la clase más votada entre los k prototipos de centroide más próximo. En problemas prácticos, al igual que ocurría con la regla de clasificación k -NN, deberíamos considerar un número impar de vecinos con el fin de evitar posibles empates (Vázquez, 2008).

2.2 Algoritmos de edición

Al iniciarse el proceso de entrenamiento de una RNA debemos contar con una muestra de entrenamiento donde la totalidad de los elementos de dicha muestra se encuentren bien etiquetados, pero en ciertas aplicaciones prácticas dicho conjunto de entrenamiento puede incluir elementos erróneamente etiquetados lo que obviamente dará lugar a una tasa de error más elevada que el valor esperado.

Si a todo esto le sumamos que en diversas ocasiones el conjunto de entrenamiento original resulta tan extremadamente grande que muchos sistemas sufren dificultades relacionadas con el tiempo de ejecución del aprendizaje, reducir el tamaño del conjunto de entrenamiento y eliminar aquellos elementos mal etiquetados nos daría una gran ventaja, puesto que aumentaríamos la velocidad de aprendizaje y se aumentaría la capacidad de generalización de la red entrenada.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Los algoritmos de edición eliminan del conjunto de entrenamiento aquellos prototipos erróneamente etiquetados y, al mismo tiempo, limpian los posibles solapamientos entre regiones de clases distintas. Estos algoritmos descartarán del conjunto de entrenamiento aquellos prototipos que no influyan explícitamente en la obtención de un resultado de clasificación igual o muy similar al obtenido, con el empleo de la totalidad del conjunto de prototipos (Mesa, 2010).

El objetivo de los métodos de edición es el de seleccionar un conjunto de referencia representativo y reducido, de forma que decrezca tanto el error de clasificación como el coste computacional de la clasificación (Nielsen, 2000).

El proceso de edición puede llevarse a cabo utilizando principalmente 3 estrategias fundamentales que se explican a continuación:

Estrategia incremental:

Se comienza a partir de un subconjunto vacío S y paso a paso se van agregando los elementos del conjunto original al subconjunto S de tal manera que estos cumplan con el criterio de selección de elementos empleado, el cual consiste en escoger el mejor elemento en $T - S$.



Figura 4 Estrategia incremental.

En esta estrategia el orden en que se presenten los elementos resulta de gran importancia debido a que existe una mayor probabilidad de ser incluido en el subconjunto S a los primeros elementos que se presenten, esto se debe a que cuando los últimos elementos se presenten existe una alta probabilidad de que ya estén representados en S por lo tanto no serán incluidos, lo que puede traer como consecuencia problemas en la

CAPÍTULO 2: MÉTODOS DE EDICIÓN

precisión si los últimos elementos representan una mayor generalización de los primeros. Debido a esto los objetos en la estrategia incremental se presentan de manera aleatoria.

Una de las principales ventajas de este tipo de estrategia consiste en que es más rápida y consume menos recursos de almacenamiento durante el proceso de entrenamiento (López, 2005).

Estrategia decremental:

En esta estrategia se comienza con el subconjunto $S = T$ y paso a paso se va eliminando de S el objeto que cumpla con el criterio de selección de objetos utilizados que en este caso consiste en eliminar el objeto que peor represente una clase en la muestra.

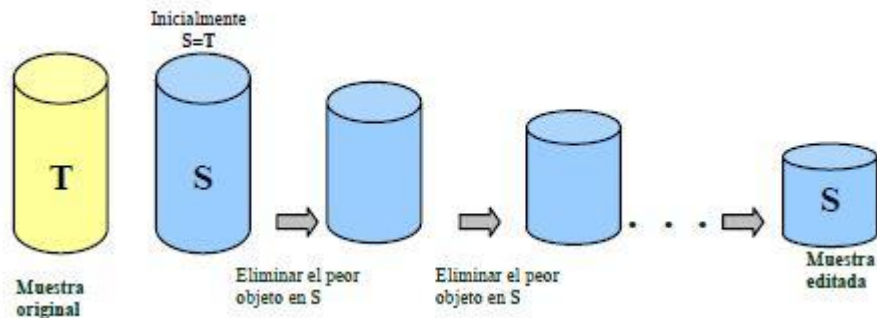


Figura 5 Estrategia decremental.

En esta estrategia también es importante el orden en que se presenten los objetos pero a diferencia de las estrategias incrementales todos los objetos almacenados están disponibles en todo momento para realizar el análisis de cuál de ellos resulta conveniente eliminar.

Una de las principales desventajas de esta estrategia es que resulta computacionalmente más costosa que las estrategias incrementales debido a que para hallar similitud entre un objeto y el subconjunto S , la estrategia decremental lleva a cabo n comparaciones (donde $n=|S|$), mientras que la estrategia incremental realiza menos cálculos (cero inicialmente y posteriormente sólo una fracción de $|T|$) (López, 2005).

Entre las principales ventajas de dicha estrategia destaca que se obtiene una mayor reducción de T y con el subconjunto S se obtiene una mayor precisión de clasificación.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Estrategia por lotes.

Esta es otra de las maneras en que puede llevarse a cabo el proceso de edición, la cual consiste en identificar y marcar aquellos objetos que no satisfacen el criterio de selección, los cuales no serán considerados en el subconjunto S y finalmente se eliminan tales objetos, es decir, no se elimina sólo un objeto sino grupos de estos. Al igual que la estrategia decremental, esta técnica resulta ser computacionalmente costosa.

2.2.1 Algoritmo de edición de Wilson.

El algoritmo de Edición de Wilson, también conocido como la regla del vecino más cercano editado (Edited Nearest Neighbor) elimina las instancias mal clasificadas de un conjunto de datos mediante la regla k -NN (Moreno, 2009).

Este algoritmo elimina la superposición entre clases y por tanto parte de posible ruido que pueda haber en la muestra. El algoritmo de ENN realiza los siguientes pasos (Moreno, 2009):

1. Recibe como parámetro el número de vecinos a buscar para cada clase.
2. Calcula los k vecinos más cercanos de cada instancia.
3. Si la mayoría de los vecinos son de clase distinta marca la instancia.
4. Elimina las instancias marcadas.

La estrategia principal del método de edición de Wilson consiste en la eliminación de los elementos que son atípicos en la muestra de entrenamiento original aplicando la técnica de clasificación de los k vecinos más cercanos la misma consiste en que dado un patrón perteneciente a la muestra original se le hayan los k vecinos más cercanos y si la clase a la que pertenece la mayoría de estos k vecinos no coincide con la del elemento en cuestión entonces es eliminado de la muestra original (Barandela, 2001).

Aún cuando este método persigue la creación de una muestra con los patrones que representen mejor a las clases, también se logra una disminución en el tamaño de la muestra de entrenamiento.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Este método resulta relativamente fácil de implementar y comprender. Por otra parte, el coste computacional para este procedimiento de edición es de $O(N^2)$, lo cual hace que para ciertos problemas prácticos donde se cuente con conjuntos de entrenamiento relativamente grandes su aplicación se puede ver limitada (Mesa, 2010).

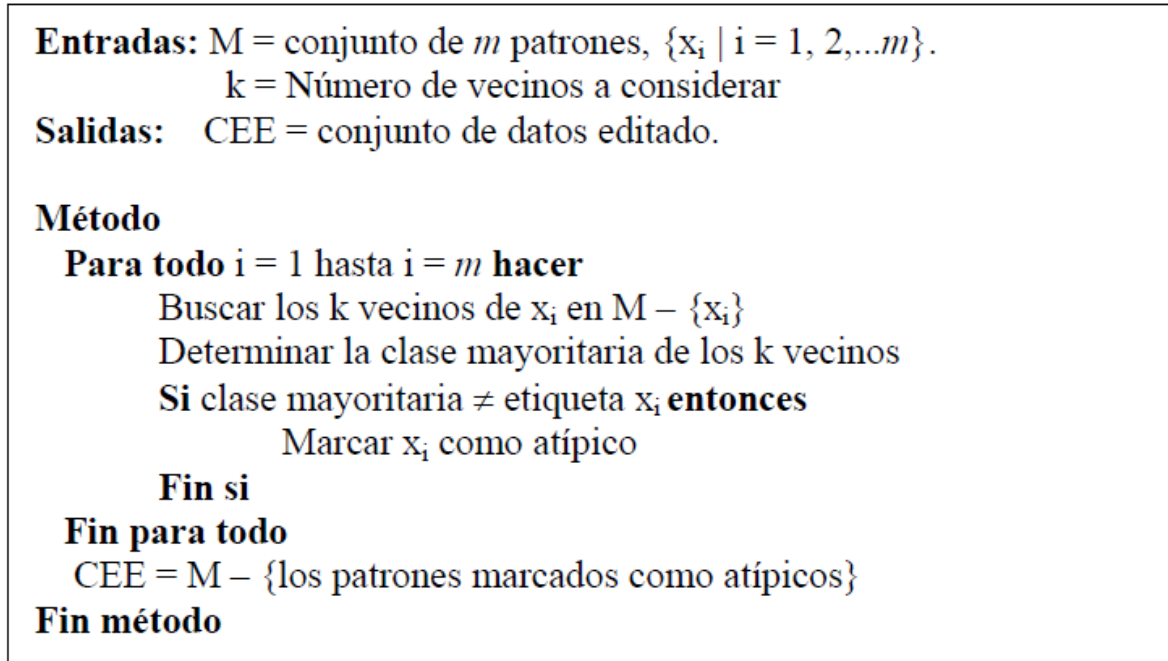


Figura 6 Algoritmo de edición de Wilson

Esta técnica es ampliamente recomendada en aquellos casos donde se encuentren clases solapadas, ya que se ha demostrado que la mayor parte de los patrones que son eliminados corresponden a aquellos patrones que se encuentran en la region de solape entre dos o mas clases.

2.2.1.1 Edicion de Wilson con distancia ponderada

Con el proposito de lograr una disminucion en los efectos negativos al aplicar el algoritmo de edicion de Wilson a un conjunto de datos que presenta desbalance, se propuso la incorporacion de una ponderacion a la distancia Euclideana, con el fin de evitar la eliminacion de patrones que puedan ser utiles, la misma se define de la siguiente forma:

$$d_w(y, x_0) = (m_i/m)^{1/n} d_E(y, x_0)$$

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Donde X_0 es un patrón que representa la clase i , m_i es el número de patrones de la clase i , m es el total de patrones de entrenamiento, d_E es la distancia Euclideana, y x es el patrón de prueba n -dimensional (Valdovinos, 2006).

2.2.2 Algoritmo de edición por Vecindad del centroide más próximo

Dicho método de edición consiste básicamente en aplicar las reglas de clasificación envolvente. Debido a que el algoritmo de Edición de Wilson obtiene buenos resultados de edición, y por otra parte los esquemas envolventes de clasificación superan a la regla K-NN, entonces se pretende lograr un algoritmo de edición que obtenga los beneficios de ambos, considerando la edición de prototipos para la regla K-NN en términos de proximidad como de distribución espacial, obteniendo de este modo, una información más precisa que nos permita ajustar las decisiones sobre los puntos frontera (Vázquez, 2008).

Sea $\{X, \Theta\} = \{(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)\}$.

Un conjunto de entrenamiento con N prototipos y M posibles clases distintas. Sea k el número de vecinos de centroide más próximo a determinar para cada prototipo. Entonces el algoritmo que recibe el nombre de Wilsoncn (k -NCN), se puede plantear de la siguiente forma (Vázquez, 2008):

Algoritmo Wilsoncn (k -NCN)

Entrada: $X \rightarrow$ Conjunto de entrenamiento a editar

$k \rightarrow$ Cantidad de vecinos

Salida: $S \rightarrow$ Conjunto editado

Método:

1- Inicialización $S \leftarrow X$

2- Para cada prototipo $x_i \in X$

2.1- Buscar los k vecinos de centroide más próximo de x_i en $X - \{x_i\}$

2.2- Si $\delta_{k\text{-NCN}}(x_i) \neq \theta_i$ entonces $S = S - \{x_i\}$

Figura 7 Algoritmo de edición del centroide más cercano.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Como puede apreciarse, este algoritmo de edición resultará tan simple y fácil de implementar como el algoritmo de edición de Wilson.

Resulta necesario también destacar que el coste computacional asociado a este método de edición, donde se calculan los k -vecinos del centroide más próximo resultará ser $O(n^2)$.

Es preciso también señalar que sobre este método se pueden realizar diversas extensiones como por ejemplo (Vázquez, 2008):

- Se puede implementar la versión repetitiva del esquema, es decir la que consiste en editar en conjunto de entrenamiento utilizando la regla k -NCN hasta que no se produzcan más eliminaciones.
- Reetiquetar determinados prototipos en función de la zona del espacio de representación que ocupe.

2.2.3 Algoritmo Backward Sequential Edition (BSE)

En un conjunto de entrenamiento X , suele ocurrir que algunos de los objetos de X no aportan información relevante para la clasificación, por lo que es necesario identificar y descartar tales objetos, es decir, realizar una edición o selección de objetos, lo cual es un problema de búsqueda que consiste en encontrar el subconjunto de objetos óptimo para el entrenamiento.

Debido a que el espacio de subconjuntos de un total de d objetos es de tamaño 2^d , los algoritmos para la selección de objetos suelen evitar emplear las técnicas exhaustivas, es decir, aquellas con las cuales se analizan las 2^d posibilidades, lo que representa un alto costo computacional, ya que éste resulta ser exponencial ($O(2^d)$) (Vázquez, 2008).

Una de las técnicas no exhaustivas es la búsqueda secuencial, cuyo orden de complejidad es polinomial ($O(d^2)$).

El método propuesto BSE es una técnica decremental no exhaustiva para la selección de objetos, el cual en cada paso descarta o elimina el objeto que menos información aporta en la calidad del subconjunto parcial. Para evaluar los subconjuntos parciales a lo largo del proceso se emplea un clasificador. La función $Classifier(P)$ devuelve como resultado el

CAPÍTULO 2: MÉTODOS DE EDICIÓN

porcentaje de clasificación correcta empleando a P como conjunto de entrenamiento (Vázquez, 2008).

A continuación se muestra el pseudocódigo de dicho método:

Algoritmo BSE (*Backward Sequential Edition*)

Entrada: $X \rightarrow$ Conjunto de entrenamiento a editar

Salida: $S \rightarrow$ Conjunto editado

Método

1- Sea $S=X$

2- $BestEval = Classifier(S)$

3- Repetir

3.1- $WorstP =$ Ninguno

3.2- Para cada objeto P en S

3.2.1- $S' = S - \{P\}$

3.2.2- Si $Classifier(S') \geq BestEval$ entonces
 $WorstP = P$

$BestEval = Classifier(S')$

3.2.3- Si $WorstP \neq$ Ninguno entonces

3.2.3- $S = S - \{WorstP\}$

4- Hasta que $WorstP ==$ Ninguno ó $S == \emptyset$

5- Devolver S

Figura 8 Algoritmo BSE.

2.2.4 Algoritmo Subconjunto Selectivo Modificado

Dicho método de reducción realiza la integración de un subconjunto de patrones más pequeño que el tamaño del conjunto original, además de garantizar una mejor

CAPÍTULO 2: MÉTODOS DE EDICIÓN

aproximación a las fronteras de decisión por mantener los patrones que se encuentran cercanos a las fronteras entre clases.

El proceso toma en cuenta los vecinos relacionados R_i de cada patrón x_i contenido en el CE, donde R_i es el conjunto de todos los y_i vecinos relacionados al patrón de entrenamiento x_i , de tal forma que y_i es de la misma clase que x_i y es más cercano a x_i que su vecino más cercano en el CE de una clase diferente (Valdovinos, 2006).

Con el propósito de ejemplificar de manera clara el método, se considera un caso de dos clases, clase 1 y clase 2. Previo a la integración de SSM se consideran los siguientes aspectos (Valdovinos, 2006):

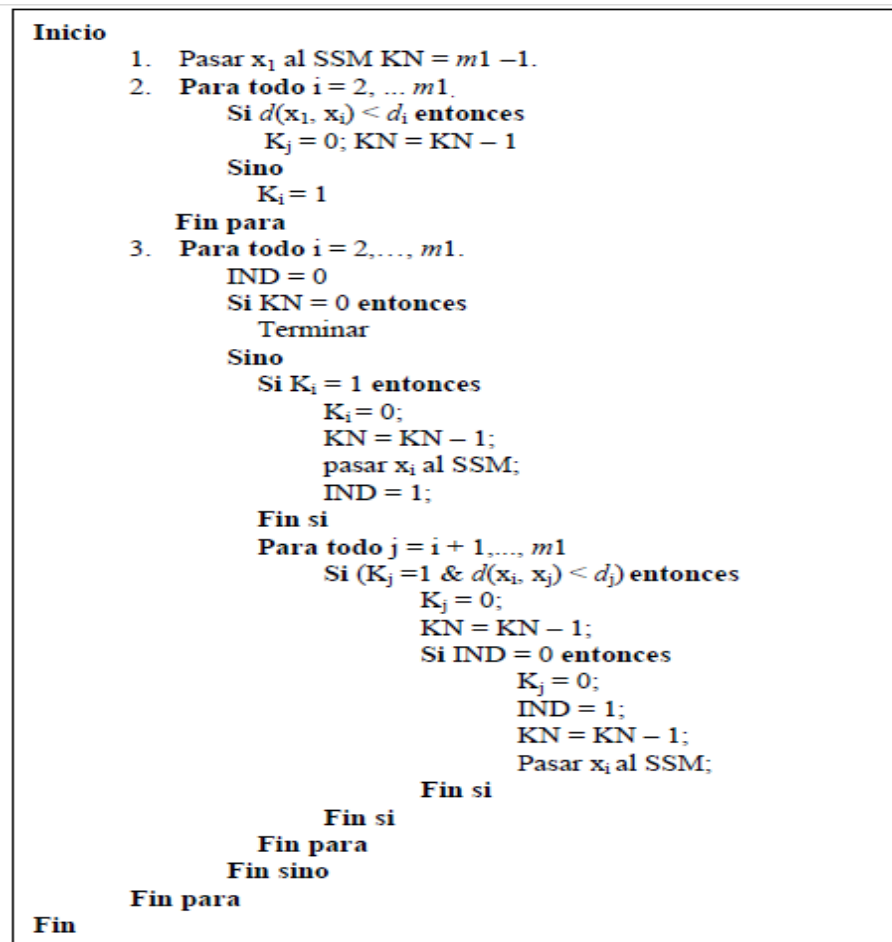


Figura 9 Algoritmo subconjunto selectivo modificado.

- m_1 es el número de patrones de la clase 1.
- IND es una bandera que impide la duplicidad de patrones en el SSM.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

- Para cada x_i , se obtiene su vecino más cercano de la clase 2 y se almacena su distancia en d_i .
- Todas las distancias son ordenadas de menor a mayor.
- KN indica el número de patrones que aun no están representados en el SSM. Cuando $KN = 0$, el algoritmo termina.

Cuando hay dos o más distancias iguales, la decisión sobre que patrón se incluirá primero en el SSM se realiza en base al patrón que resulte ser vecino relacionado para un número mayor de patrones de entrenamiento y, en caso que esta situación también sea de empate, entonces la selección se realiza de manera aleatoria (Valdovinos, 2006).

Esta condición hace al algoritmo un poco más complejo desde el punto de vista computacional, pero asegura la singularidad de la solución y también tendrá la ventaja de propiciar una mayor reducción del tamaño del SSM.

2.2.5 Algoritmo de edición basado en Teoría de Conjuntos Aproximados

La Teoría de Conjuntos Aproximados está basada en aproximar cualquier concepto, un subconjunto duro del dominio, como por ejemplo una clase en un problema de clasificación supervisada, por un par de conjuntos exactos, llamados aproximación superior y aproximación inferior del concepto (Caballero, 2007).

Con esta teoría es posible tratar tanto datos cuantitativos como cualitativos, y no se requiere eliminar las inconsistencias previas al análisis. La inconsistencia describe una situación en la cual hay dos o más variables en conflicto para ser asignados a una variable.

2.2.5.1 Consideraciones necesarias sobre la Teoría de Conjuntos Aproximados

Los conceptos básicos de la Teoría de Conjuntos Aproximados (TCA) son las aproximaciones inferiores y superiores de un subconjunto $X \subseteq U$. Estos conceptos fueron originalmente introducidos con referencia a una relación de inseparabilidad R . Sea R una relación binaria definida sobre U la cual representa la inseparabilidad, se dice que $R(x)$ significa el conjunto de objetos los cuales son inseparables de x (Caballero, 2007).

Así, $R(x) = \{y \in U: yRx\}$.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

En la TCA clásica, R es definida como una relación de equivalencia; es decir, es una relación binaria $R \subseteq U \times U$ que es reflexiva, simétrica y transitiva. R induce una partición de U en clases de equivalencia correspondiente a $R(x)$, $x \in U$ (Caballero, 2007).

El objetivo es construir una relación R' a partir de la relación de inseparabilidad R pero flexibilizando las condiciones originales para la inseparabilidad.

Esta flexibilización puede ser realizada de múltiples formas, así como pueden ser dadas varias definiciones posibles de similitud. Existen varias funciones de comparación de atributos (funciones de similitud), las cuales están asociadas al tipo del atributo que se compara.

La aproximación de un conjunto $X \subseteq U$, usando una relación de inseparabilidad R, ha sido inducida como un par de conjuntos llamados aproximaciones R-inferior y R-superior de X.

Se considera en esta propuesta una definición de aproximaciones más general, la cual maneja cualquier relación reflexiva R'. Las aproximaciones R-inferior ($R_*(X)$) y R-superior ($R^*(X)$) de X están definidas respectivamente como se muestra (Caballero, 2007):

$$R_*(X) = \{x \in X : R(x) \subseteq X\}$$
$$R^*(X) = \bigcup_{x \in X} R(x)$$

Figura 10 Fórmulas R-superior y R- inferior.

2.2.5.2 Función de pertenencia aproximada

Esta función cuantifica el grado de solapamiento relativo entre $R'(x)$ (clase de similitud de x) y la clase a la cual el objeto x pertenece. Se define como sigue (Caballero, 2007):

$$\mu_X(x) = \frac{|X \cap R(x)|}{|R(x)|}$$

Figura 11 Función de pertenencia aproximada.

CAPÍTULO 2: MÉTODOS DE EDICIÓN

La función de pertenencia aproximada puede ser interpretada como una estimación basada en frecuencias de $\Pr(x \in X | x, R'(x))$, es decir, la probabilidad condicional de que el objeto x pertenezca al conjunto X .

2.2.5.3 Algoritmo EditBRS

A continuación se describen los pasos para la aplicación del método EditBRS (Caballero, 2007).

1. Construir el conjunto B , $B \subseteq A$. Se sugiere que B sea un reducto del sistema de decisión para disminuir la dimensionalidad de los atributos sin afectar el proceso de clasificación.
2. Formar los conjuntos $X_i \subseteq U$, tal que todos los elementos del universo (U) que tienen valores d_i en el atributo de decisión están en X_i .
3. $S_E = \Phi$.
4. Para cada conjunto X_i calcular su aproximación inferior y su aproximación superior.
 - 4.1. $S_E = S_E \cup B^*(X_i)$.
 - 4.2. $T_i = B^*(X_i) - B_*(X_i)$.
5. Calcular la unión de los conjuntos T_i , para obtener $T = \cup T_i$.
6. Para cada elemento del conjunto T :
 - 6.1 Calcular la función de pertenencia aproximada a cada clase.
 - 6.2 Si la clase con mayor valor (según 6.1) difiere de la clase del objeto, entonces reetiquetarla por esta clase con la cual se obtuvo mayor valor de pertenencia del objeto.
7. $S_E = S_E \cup T$. El conjunto de entrenamiento editado se obtiene como el conjunto resultante en S_E .

2.2.6 Algoritmo de edición basado en probabilidad de clases

El algoritmo de edición basado en probabilidad de clases utiliza una nueva regla de clasificación la cual propone tener siempre en cuenta la probabilidad que tiene un objeto de pertenecer a una clase determinada (Mesa, 2010).

CAPÍTULO 2: MÉTODOS DE EDICIÓN

Con el fin propuesto se definió la siguiente ecuación siempre positiva (Mesa, 2010):

$$P_i(x) = \sum_{j=1}^k P_i^j \frac{1}{(1 + d(x, x^j))}$$

Figura 12 Ecuación probabilidad de clases.

En la misma p_i^j expresa la probabilidad de que el j -ésimo vecino más cercano x^j pertenezca a la clase i .

Resulta necesario destacar que la probabilidad de un objeto de pertenecer a su clase es 1 y para cada una de las restantes clases la probabilidad de pertenencia a las mismas es 0.

Este método de edición de manera general compara la clase a la que mayor probabilidad de pertenecer tiene un objeto dado con la clase a la que tiene asignado que pertenezca y si no coinciden el elemento en cuestión es eliminado.

Cabe destacar que de utilizarse un solo vecino más cercano el comportamiento del algoritmo es similar al algoritmo de edición de Wilson.

A continuación se presenta el pseudocódigo del método (Mesa, 2010):

1-Inicialización: $S = X$.

2- Para cada prototipo $x \in X$

2.1- Buscar los k vecinos más cercanos en $X - \{x\}$.

2.2- Si $\mathcal{Z}_{k\text{-prob}} \neq \theta$, hacer $S = S - \{x\}$, donde θ denota la clase a la que pertenece el objeto.

2.2.7 Algoritmo de edición por partición.

En este esquema de edición, el método de estimación de la pertenencia de un prototipo a su clase consistirá, en realizar una partición del conjunto de entrenamiento en m bloques

CAPÍTULO 2: MÉTODOS DE EDICIÓN

disjuntos de prototipos y, después de enumerarlos, hacer una estimación para cada bloque j utilizando el bloque $((j+1) \bmod m)$ para diseñar el clasificador (Mesa, 2010).

Este procedimiento se puede considerar estadísticamente independiente siempre que $m > 2$ (Mesa, 2010).

A continuación se presenta el pseudocódigo del método (Mesa, 2010):

- 1- Hacer una partición aleatoria de X en m bloques.
- 2- Para cada bloque T_j .
 - 2.1- Para cada $x_i \in T_j$.
 - 2.1.1- Buscar los k vecinos más cercanos de x_i en T_{j+1} .
 - 2.1.2- - Si $\mathcal{O}_{k\text{-NN}}(x_i) = \theta_i$ hacer $X = X - \{x_i\}$.

CONCLUSIONES DEL CAPÍTULO

En este capítulo se exponen detalladamente los métodos de edición del conjunto de datos de entrenamiento de las RNA, los cuales son utilizados para eliminar del conjunto de datos original aquellos prototipos erróneamente etiquetados, lo que conlleva a una mejora en la capacidad de generalización de la red entrenada, así como a un aumento de la velocidad de aprendizaje de la red.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

Introducción

En el presente capítulo se exponen los resultados obtenidos de la aplicación de diferentes métodos de edición expuestos en el capítulo anterior, con las modificaciones realizadas a los mismos.

3.1 Comparación teórica de los métodos de edición

Teniendo en cuenta lo expuesto en el capítulo anterior se realiza una comparación entre los métodos de edición del conjunto de datos de entrenamiento de las RNA en cuanto a los aspectos teóricos que se mencionan a continuación:

- Curva de aprendizaje: Expresa la facilidad de comprensión que posee el algoritmo, es decir, cuan fácil es de comprender en función del tiempo que se ocupa para dicho aprendizaje.
- Complejidad de implementación: Se refiere a la complejidad propiamente dicha del algoritmo en su implementación.
- Complejidad algorítmica: Complejidad inherente a la resolución de un problema computable.

Tabla 1 Comparación teórica de los métodos de edición.

Métodos	Términos comparativos		
	Complejidad algorítmica	Curva de aprendizaje	Complejidad de implementación
Wilson	$O(n^2)$	Baja	Baja
WilsonNcN	$O(n^2)$	Media	Media
Probabilidad de clases	$O(n^2)$	Media	Media
BSE	$O(n^3)$	Media	Alta

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

SSM	$O(n^2)$	Media	Media
EditBRS	$O(n^2)$	Alta	Alta

Como se refleja en la tabla anterior la mayoría de los algoritmos de edición poseen una facilidad de comprensión e implementación media, y de manera general su complejidad algorítmica es del orden $O(n^2)$, lo que facilita su uso y eficiencia.

3.2 Modificaciones a los métodos de edición.

La mayoría de los métodos de edición estudiados en el capítulo anterior presentan una desventaja común, la misma está relacionada a que son completamente dependientes de que les sea especificado claramente la clase a la que pertenece el objeto, con la clase nos referimos al valor de salida de la red, y que este dada por un solo valor, es decir se está obligado a que la red posea una sola variable de salida, lo que en la práctica no se cumple siempre. Por lo que se hizo necesaria la modificación de dichos métodos de edición para lograr utilizarlos en cualquier conjunto de datos de entrenamiento.

Dicha modificación consiste en permitir que la clase venga dada no solo por una variable sino por varias al mismo tiempo permitiendo que la red pueda tener varias variables de salida, para esto fue necesario introducir un nuevo concepto: proximidad, el cual hace referencia a la cercanía que existe entre dos clases, es decir ya no se vería como una igualdad entre clases para la aplicación de los métodos de edición sino como que una clase es próxima a la otra.

Para ello fue utilizado un rango de proximidad que determina cuando una clase es próxima a la otra, es decir si la distancia entre una clase y otra no sobrepasa el valor del rango de proximidad se dice que esas clases son próximas.

Teniendo en cuenta estos aspectos todos los métodos de edición fueron modificados en su implementación y los resultados que se obtuvieron con los mismos son expuestos en este capítulo, con el objetivo de mostrar el desempeño de los mismos.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

3.1 Editor del conjunto de entrenamiento de las RNA

Teniendo en cuenta las modificaciones realizadas a los métodos de edición explicadas en el epígrafe anterior se realizó una aplicación que implementa varios de los métodos de edición expuestos.

La misma fue utilizada para obtener los resultados expuestos en este capítulo.

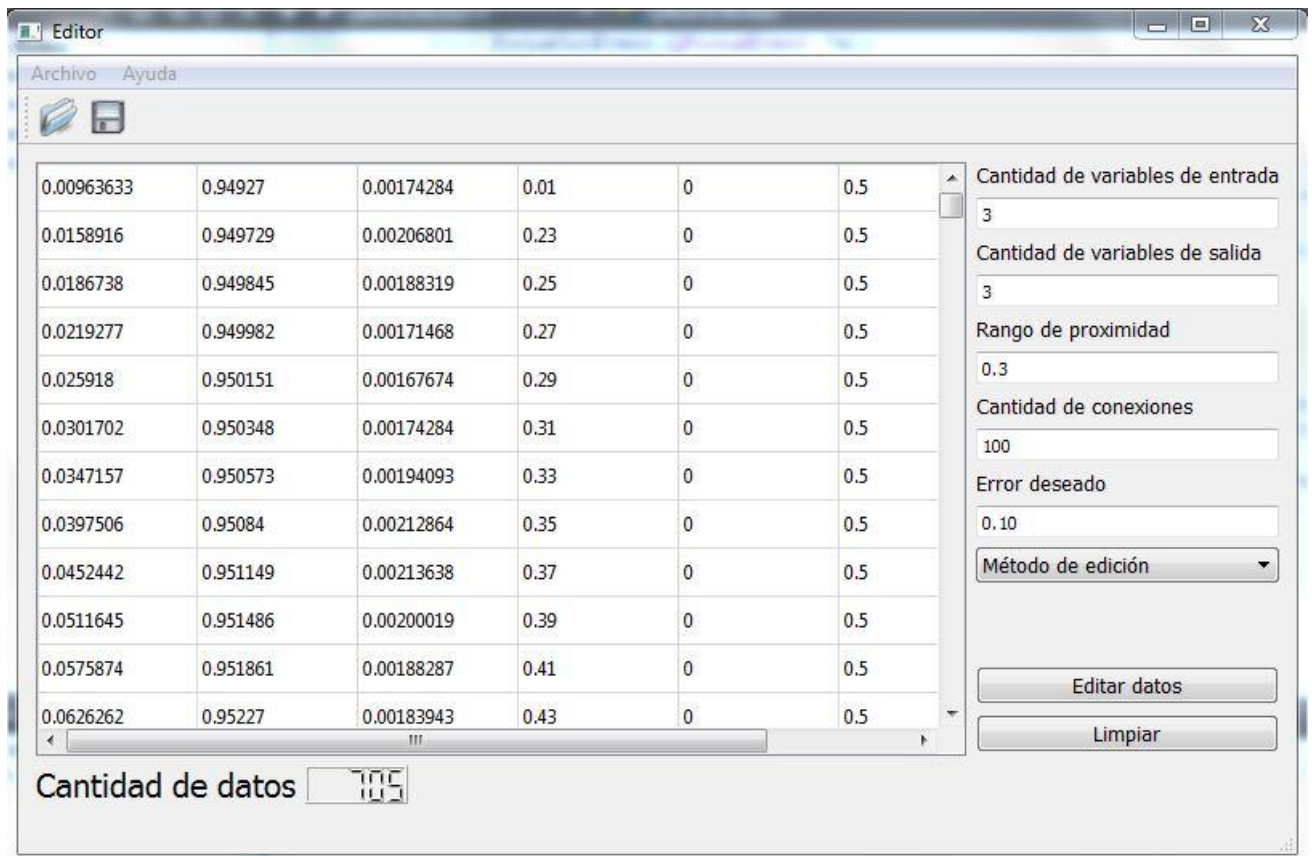


Figura 13 Editor del conjunto de entrenamiento

Dicha aplicación es capaz de cargar los datos de entrenamiento, editarlos y posteriormente guardar los datos editados, recibe los siguientes datos:

- Cantidad de variables de entrada: Se refiere a la cantidad de variables de entrada que tiene el conjunto de entrenamiento.
- Cantidad de variables de salida: Se refiere a la cantidad de variables de salida que tiene el conjunto de entrenamiento.
- Rango de proximidad: Hace referencia al rango de proximidad entre una clase y otra.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

- Cantidad de conexiones: Se refiere a la cantidad de conexiones entre neuronas que posee la RNA para la cual se están editando los datos.
- Error deseado: Se refiere al error deseado en el entrenamiento de la red.

Tanto la cantidad de variables de entrada y salida como el rango de proximidad son utilizadas por los métodos de edición para la realización de los mismos. En tanto la cantidad de conexiones y el error deseado, se utilizan para determinar la cantidad de datos de entrenamiento que se recomienda para el aprendizaje de la red en cuestión siendo determinado por la siguiente fórmula:

$$Cd = Cc/E$$

Donde Cd representa la cantidad de datos de entrenamiento recomendable, Cc la cantidad de conexiones de la red y E el error deseado con el cual se desea que se entrene la red.

3.3 Resultados obtenidos por cada uno de los métodos de edición modificados en cuanto a reducción de la muestra de entrenamiento

A continuación se muestran los resultados obtenidos utilizando algunos de los algoritmos de edición del conjunto de datos de entrenamiento de las RNA presentados en el capítulo anterior utilizando las modificaciones realizadas a los mismos. Para ello fueron utilizadas algunas de las bases de datos presentes en el repositorio UCI Machine Learning Database Repository.

Después de la aplicación de dichos métodos se tomaron los resultados arrojados para obtener la reducción del conjunto de datos de entrenamiento de cada uno de los métodos.

En las figuras siguientes se muestran los resultados experimentales utilizando los algoritmos de edición: Wilson Modificado, Probabilidad de Clases Modificado, por Particiones Modificado, sobre las bases de datos: Cáncer, Diabetes, Hebberman Survival y Mammographic Mass, respectivamente.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

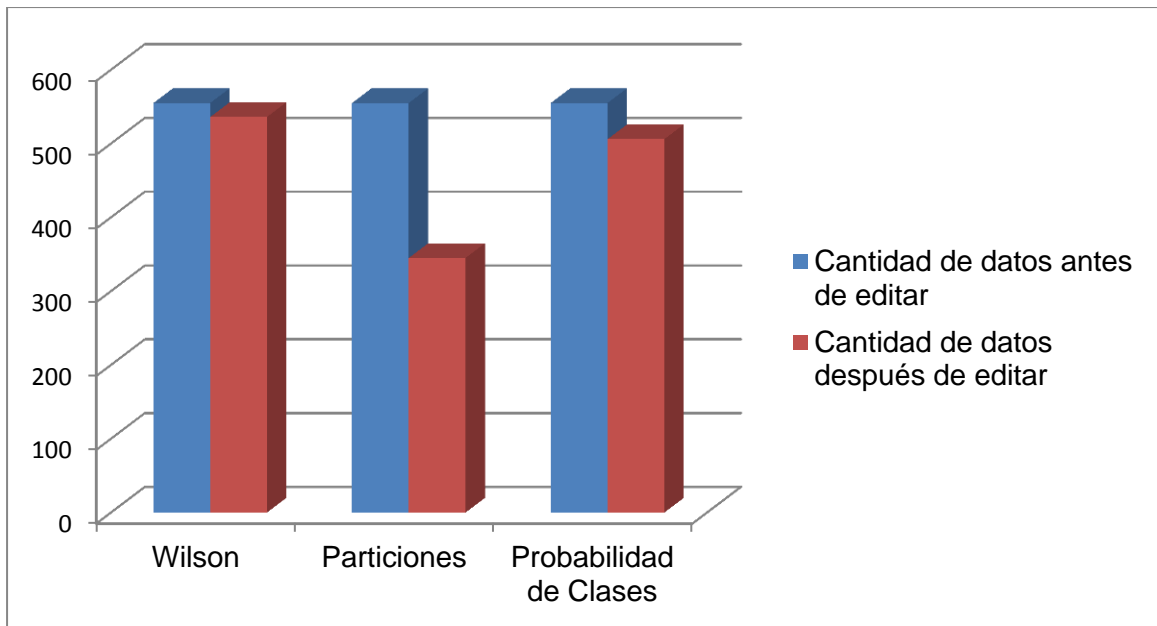


Figura 14 Reducción del tamaño de la muestra de la base de datos Cáncer.

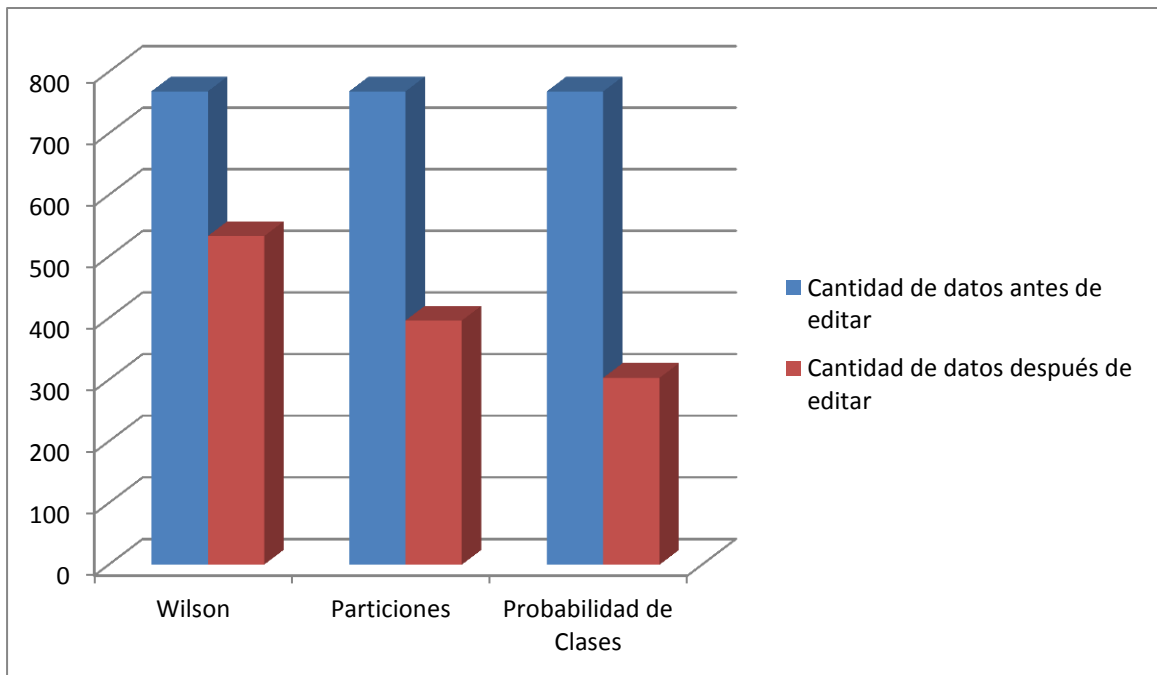


Figura 15 Reducción del tamaño de la muestra de la base de datos Diabetes.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

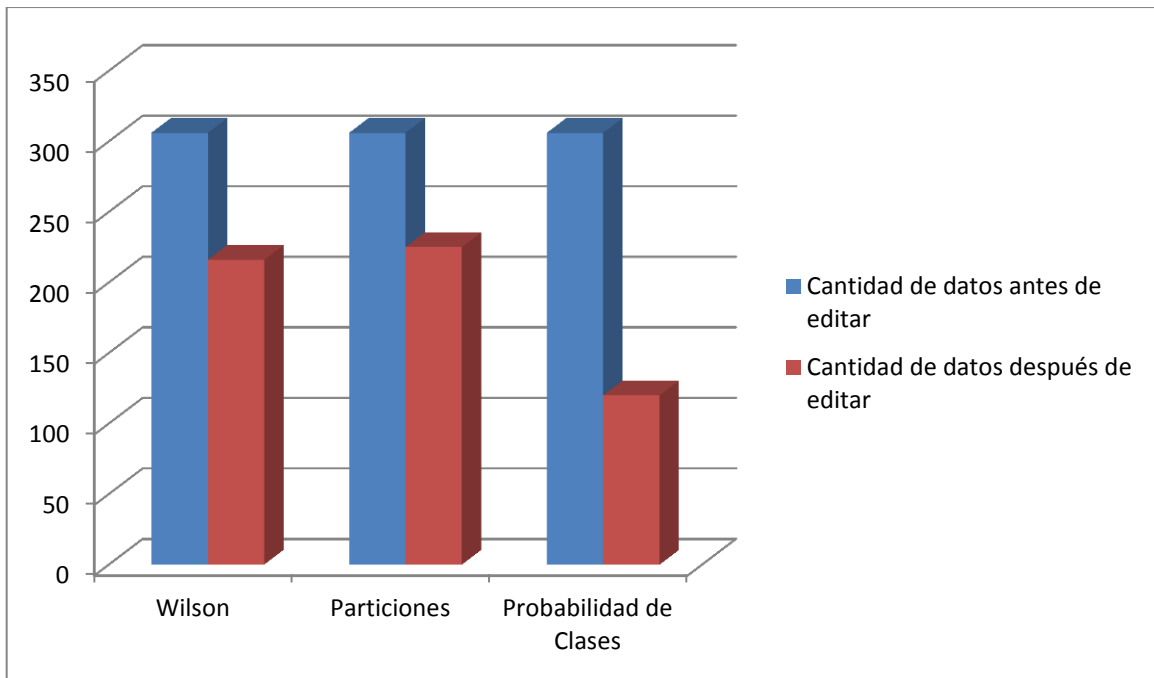


Figura 16 Reducción del tamaño de la muestra de la base de datos Hebberman Survival.

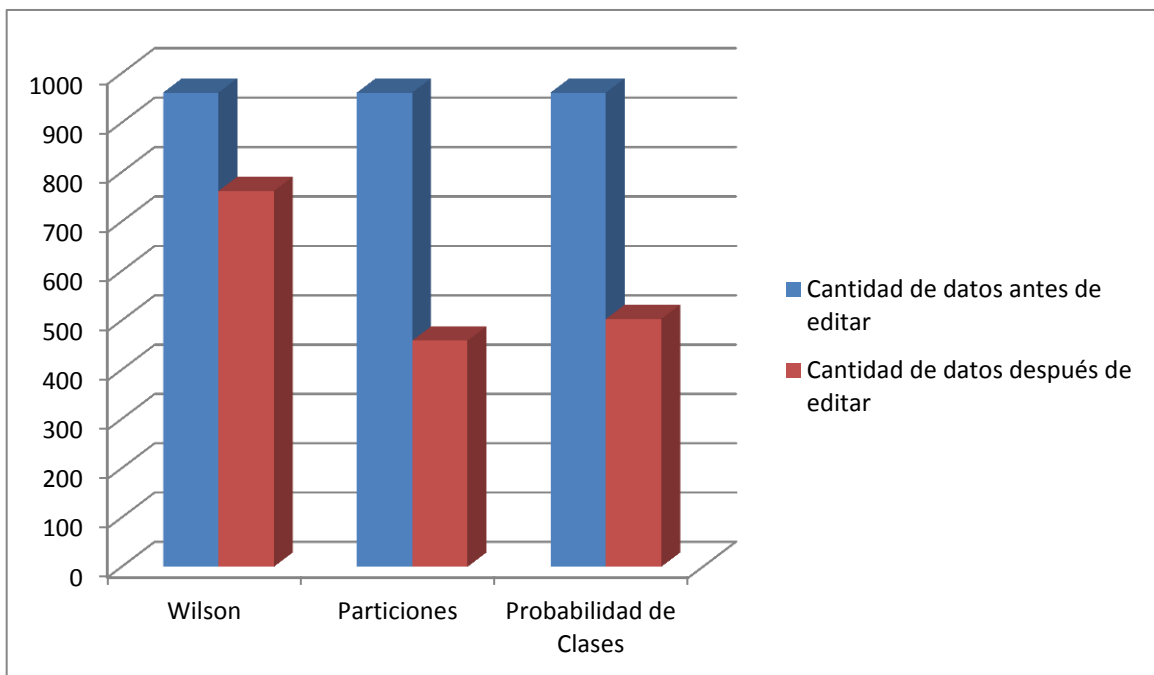


Figura 17 Reducción del tamaño de la muestra de la base de datos mammographic mass.

Uno de los aspectos de mayor interés a la hora de comparar el desempeño de los métodos de edición es la reducción del tamaño de los datos de entrenamiento después de haber sido editado el conjunto original, lo que se traduce en una disminución de la carga computacional en la fase de entrenamiento de la red.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

Como se evidencia en las gráficas anteriores los métodos de edición después de modificados fueron capaces de disminuir el tamaño de la muestra de entrenamiento en la mayoría de los casos, el método con el cual se obtiene la mayor reducción de los datos de manera general es el método edición por Particiones.

Es importante aclarar que la reducción del tamaño de los datos, no necesariamente se traduce en una mejora en el proceso de generalización por lo que debemos ser cuidadosos a la hora de escoger el método edición deseado para cierto conjunto de datos, se deben tener en cuenta tanto la reducción del tamaño de la muestra como la capacidad de generalización de la misma.

3.4 Resultados obtenidos por cada uno de los métodos de edición modificados en cuanto a disminución del error durante el entrenamiento

Uno de objetivos fundamentales de aplicar los métodos de edición sobre un conjunto de datos de entrenamiento consiste en mejorar la capacidad de generalización de la red, o sea, disminuir el error obtenido durante el proceso de entrenamiento de la RNA. A continuación se muestran los resultados obtenidos de la comparaciones realizadas en el entrenamiento a una RNA utilizando conjuntos de datos editados y sin editar para ello fueron utilizadas las bases de datos cáncer y mammographic mass ambas bases de datos reales tomadas de UCI Machine Learning Database Repository.

Para la obtención de los resultados del error durante el entrenamiento fue utilizado el editor de redes neuronales artificiales que posee el proyecto Navegación y Comportamiento Inteligente.

En el mismo fue utilizado el motor de RNA FANN, las funciones de activación para cada una de las capas fue la sigmoïdal logarítmica, un impulso de aprendizaje de 0.10, y la función de entrenamiento que se utilizo fue RDROP.

Es importante aclarar que todos los métodos de edición fueron probados bajo las mismas condiciones.

Los métodos de edición utilizados fueron Wilson Modificado, Particiones Modificado y Probabilidad de Clases Modificado, las gráficas presentadas contienen la imagen del error obtenido con la muestra sin editar a la izquierda y con la muestra editada a la derecha.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

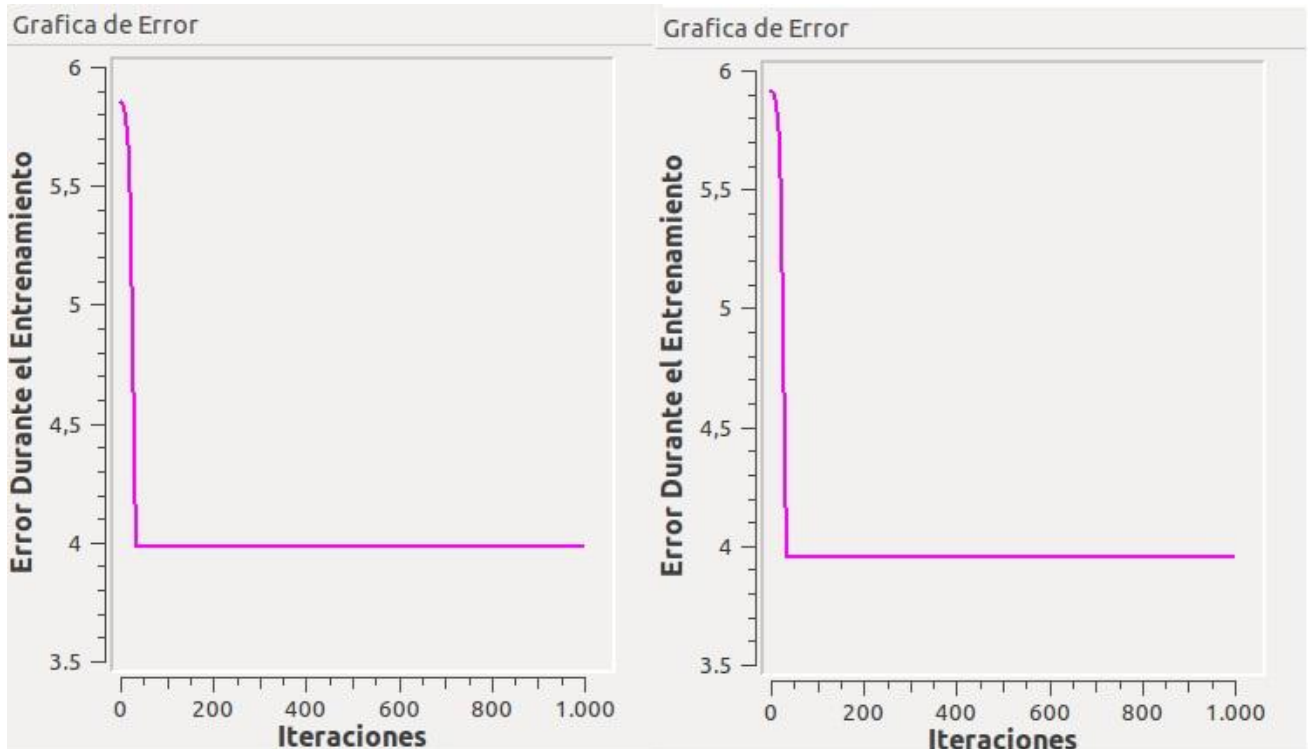


Figura 18 Gráfica del error utilizando el algoritmo de edición de Wilson sobre el conjunto de entrenamiento cáncer

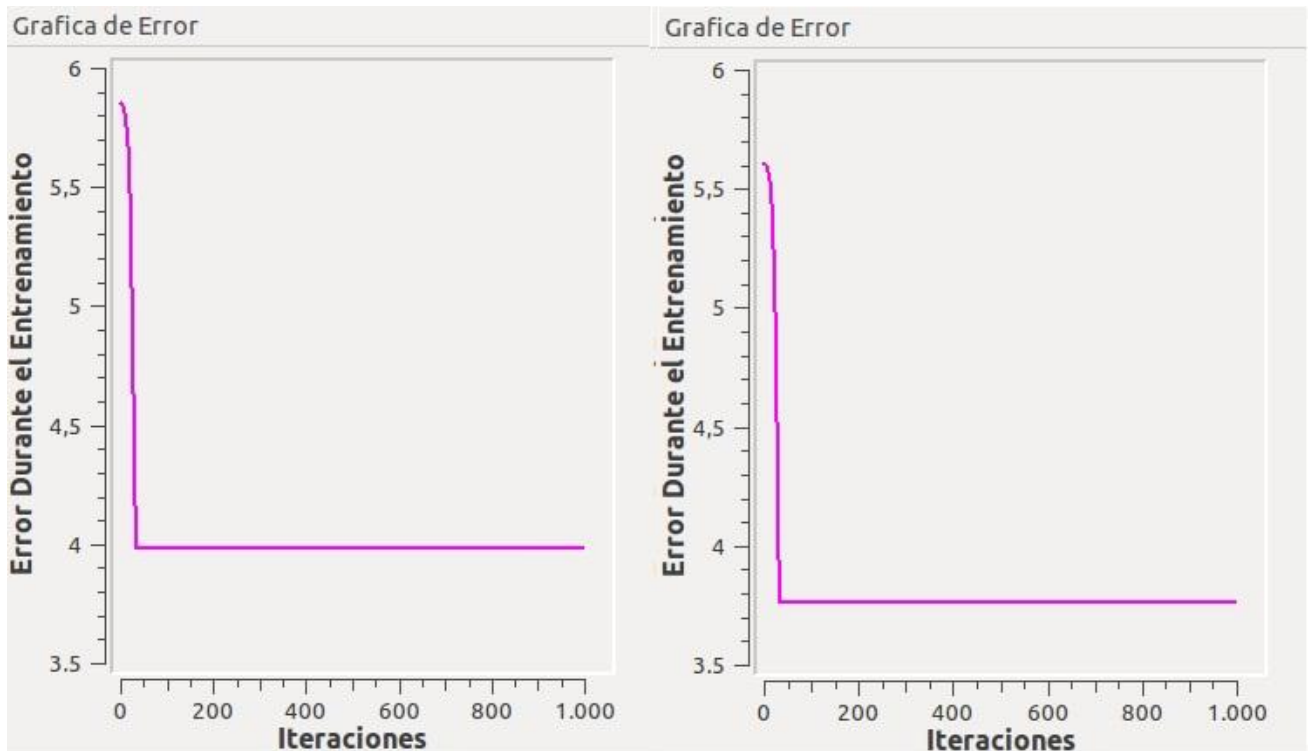


Figura 19 Gráfica del error utilizando el algoritmo de edición por Particiones sobre el conjunto de entrenamiento cáncer.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

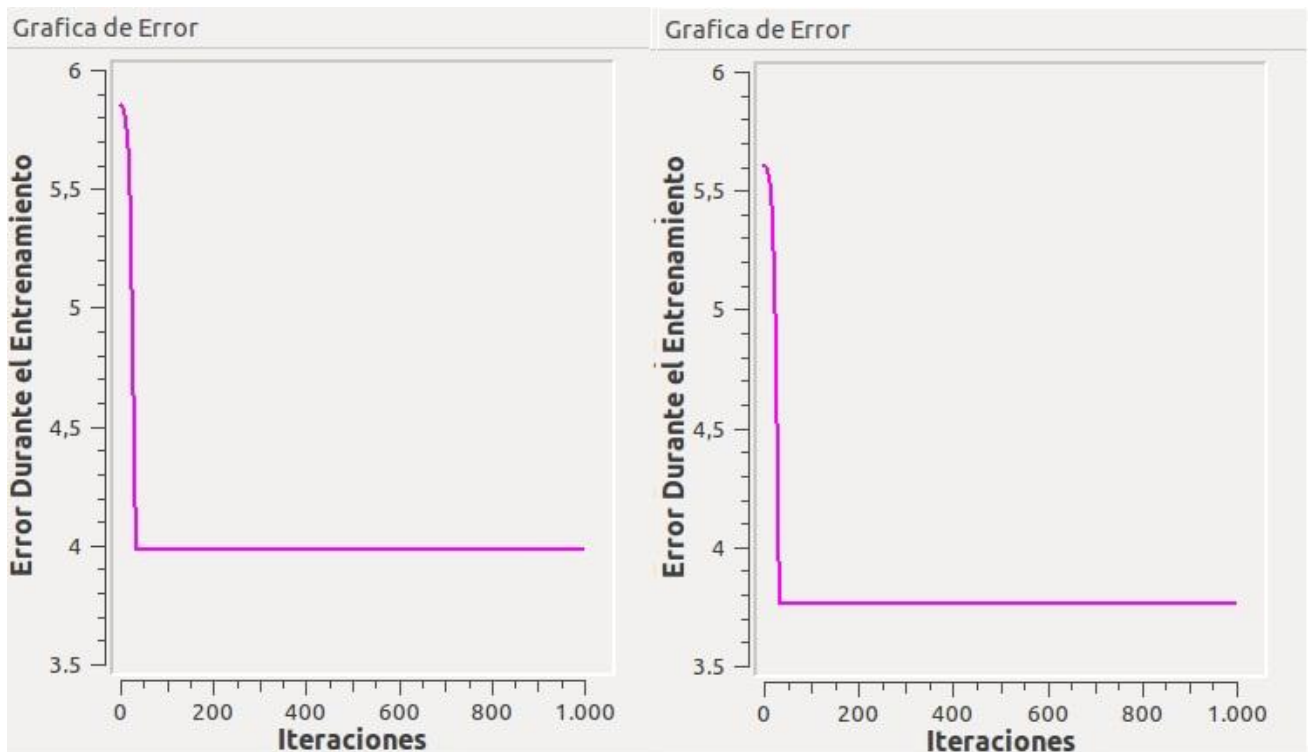


Figura 20 Gráfica del error utilizando el algoritmo de edición por Probabilidad de clases sobre el conjunto de entrenamiento cáncer.

Las figuras 12, 13 y 14 muestran como se redujo el error durante el entrenamiento si utilizamos los métodos de edición modificados sobre el conjunto de entrenamiento original cáncer.

Utilizando la base de datos sin editar el error que nos brindó la red fue de 3.98, después de la aplicación del método de edición de Wilson el error fue de 3.95, con el método por Particiones el error brindado por la red fue de 2.57, y con el método por Probabilidad de Clases fue de 3.75.

El método que nos brindo una mayor disminución del error fue el método de edición por Particiones.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

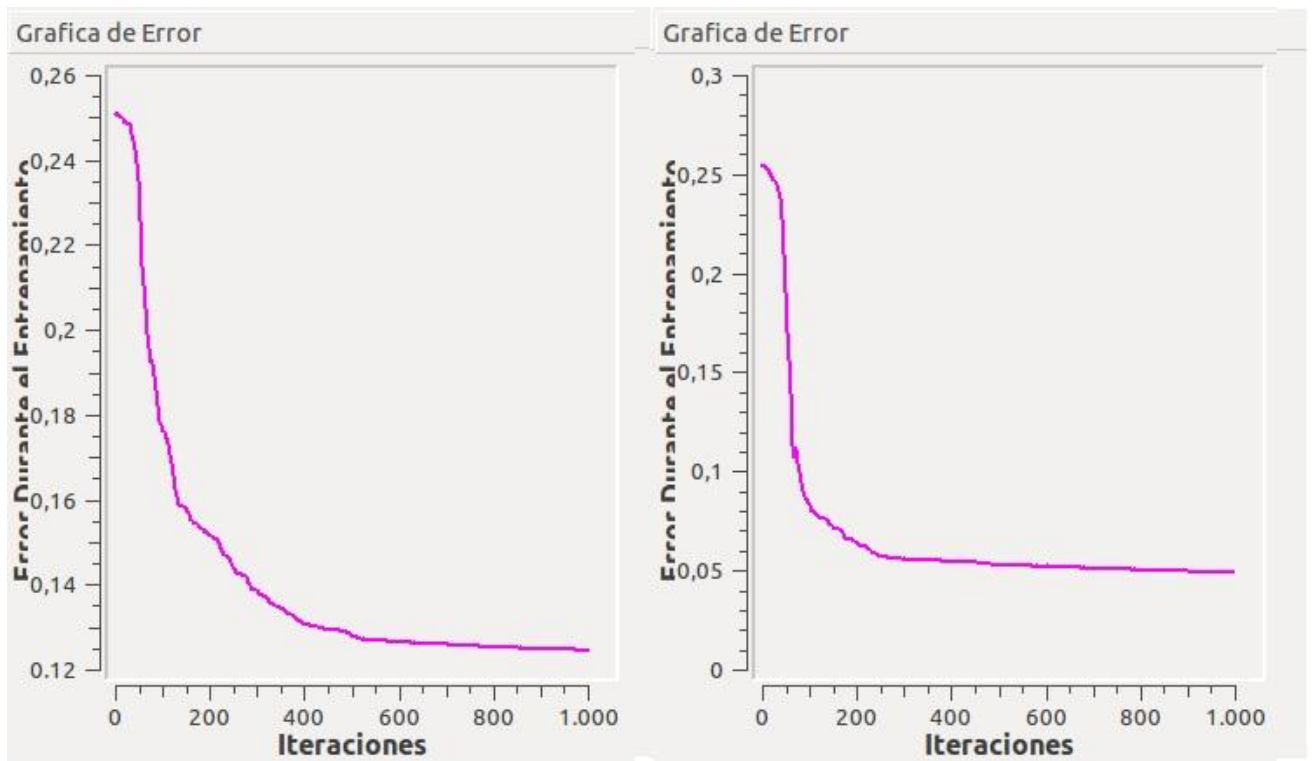


Figura 21 Gráfica del error utilizando el algoritmo de edición de Wilson sobre el conjunto de entrenamiento mammographic mass.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

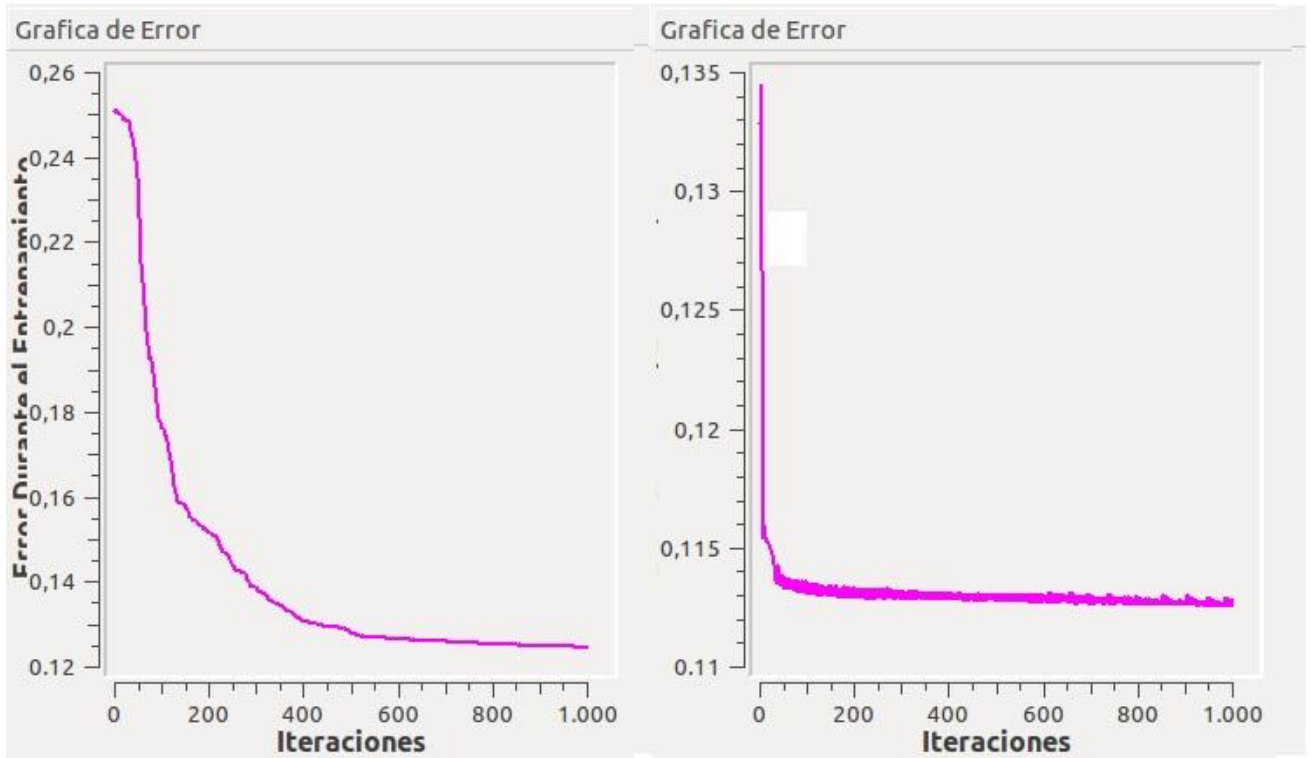


Figura 22 Gráfica del error utilizando el algoritmo de edición por Particiones sobre el conjunto de entrenamiento mammographic mass.

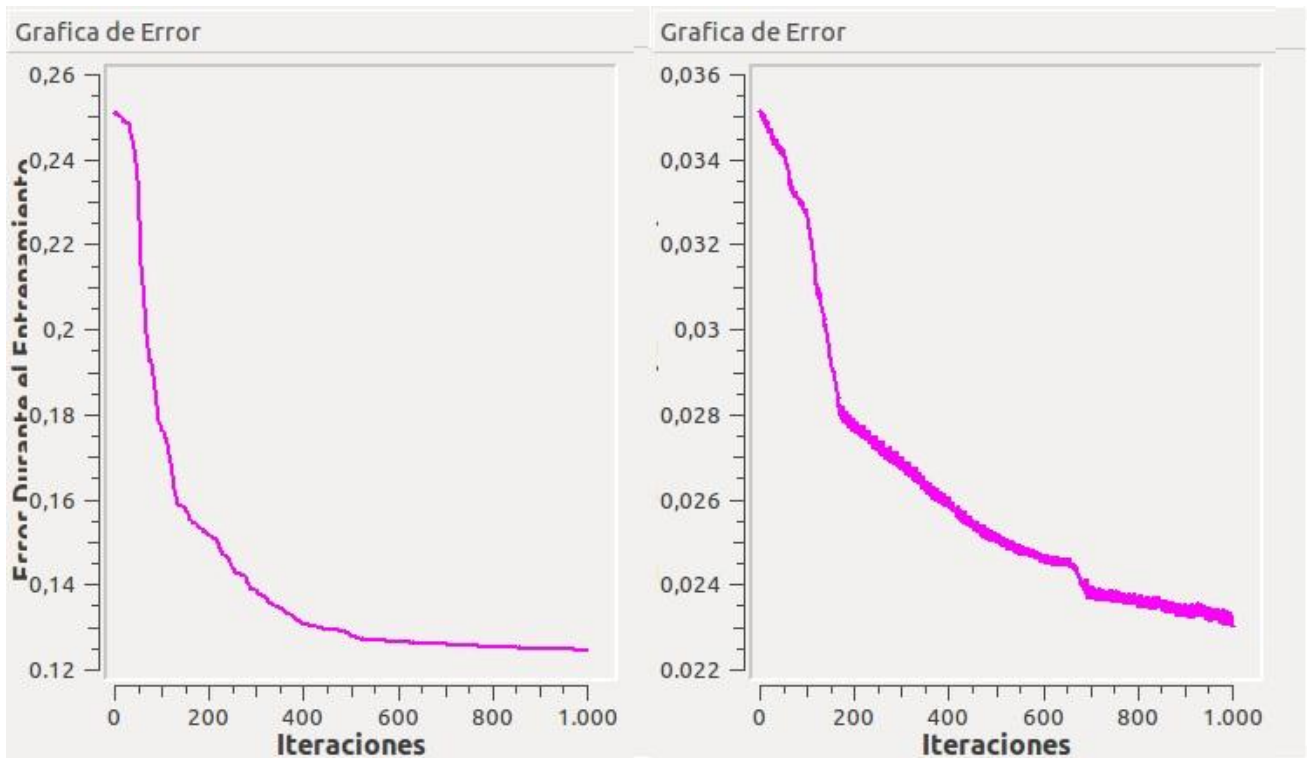


Figura 23 Gráfica del error utilizando el algoritmo de edición por Probabilidad de clases sobre el conjunto de entrenamiento mammographic mass.

CAPÍTULO 3: RESULTADOS EXPERIMENTALES

Las figuras 15, 16 y 17 también nos muestran una disminución en el error generado durante el entrenamiento con la utilización de los métodos de edición modificados.

El error que brindó la red con la muestra sin editar fue de 0.12, después de la aplicación del método de edición de Wilson el error fue de 0.06, con el método por Particiones fue de 0.11 y con el método por Probabilidad de Clases el error generado fue de 0.02.

En este caso el método que nos brinda una mayor discusión del error fue el método de edición utilizando Probabilidad de clases.

Todo esto demuestra la eficacia de los métodos de edición después de aplicada la modificación en los mismos además de la importancia de la utilización de los métodos de edición sobre un determinado conjunto de entrenamiento para lograr un aumento en la capacidad de generalización de la red.

CONCLUSIONES DEL CAPÍTULO

En este capítulo se realizó una comparación entre la eficiencia de los diferentes métodos de edición modificados demostrando las posibilidades reales que nos brindan en su uso, además de que la eficiencia del método depende de 2 factores en conjunto la reducción de la talla de la muestra de entrenamiento y la capacidad de generalización de la red.

También se realizó una comparación entre el error obtenido en el proceso de entrenamiento utilizando un conjunto de entrenamiento sin editar y uno después de habersele aplicado los métodos de edición lo que demostró la importancia de la utilización de los métodos de edición para lograr una disminución del error durante el entrenamiento.

CONCLUSIONES GENERALES

En la presente investigación se realizó un estudio relacionado con los diferentes métodos de edición del conjunto de datos de entrenamiento la cual arrojó las siguientes conclusiones:

- Se realizó una modificación a los métodos de edición existentes, la cual arrojó excelentes resultados en cuanto a la reducción del tamaño de la muestra de entrenamiento y al error de generalización de la red después de entrenada.
- Se realizó una comparación entre el comportamiento de los métodos de edición donde se comprobó que para medir la eficiencia de un método de edición sobre cierto conjunto de datos es necesario tener en cuenta los aspectos: reducción de la talla de la muestra y capacidad de generalización de la red después de entrenada con los datos editados.

RECOMENDACIONES

Dadas las conclusiones a las que se llegó con el presente trabajo recomendamos:

- Implementar los demás métodos de edición del conjunto de datos de entrenamiento aplicando las modificaciones.
- Incorporar la opción de edición del conjunto de datos de entrenamiento al editor de RNA que posee el proyecto Navegación y Comportamiento Inteligente.

BIBLIOGRAFÍA

- Artificial, Inteligencia. 2006.** Inteligencia Artificial. [Online] 2006. [Cited: 11 13, 2010.] http://www.secyt.frba.utn.edu.ar/gia/inteligencia_artificial.htm.
- Barandela, Ricardo. 2001.** *Corrección de la muestra para el aprendizaje del Perceptron Multicapa*. España : s.n., 2001.
- C++.** 2011. [Online] 2011. [Cited: 2 13, 2011.] <http://es.scribd.com/doc/54149231/46/3-2-4-ventajas-de-C>.
- Caballero, Yaile. 2007.** *La Teoría de los Conjuntos Aproximados y las Técnicas de Bootstrap para la Edición de Conjuntos de Entrenamiento*. 2007.
- Document. 2010.** Qt Reference Documentation. [Online] 2010. [Cited: 12 10, 2010.] <http://doc.trolltech.org/4.4/index.html>.
- Elaine Rich, Kevin Knight. 1991.** *Artificial Intelligence*. 1991.
- Fernández, Luis Alberto García. 2004.** [Online] 2004. <http://www.uv.mx/cienciahombre/revistae/vol17num3/articulos/inteligencia/index.htm>.
- Fuente, María Jesús de la. 2009.** *Ventajas de las RNA*. [.ppt] Valladolid : s.n., 2009.
- González, Manuel F. 2003.** *Aprendizaje y entrenamiento de RNA*. 2003.
- Graells, Dr. Pere Marques. 2008.** Grandes aportes de las TIC. [Online] 2008. [Cited: 10 22, 2010.] <http://www.peremarques.net>.
- Guzman, Maria Victoria. 2001.** Laboratorio de Dinamica no Lineal. [Online] 2001. [Cited: 10 26, 2010.] <http://www.dynamics.unam.edu/DinamicaNoLineal/Articulos/MineriaDatos/Articulo03.pdf>.
- Hetch, Nielsen R. 1988.** *Neurocomputing*. EE.UU : s.n., 1988.
- Kurzweil, Ray. 1990.** *The age of intelligent machines*. 1990.
- López, Arturo Olvera. 2005.** *Edición de muestras basado en búsqueda secuencial*. Mexico : s.n., 2005.
- Mesa, Vazquez. 2010.** *Una propuesta basada en la estimación de las probabilidades para la edición utilizando el clasificador k-NN*. [pdf] Santiago de Cuba : s.n., 2010.
- Moreno, J. 2009.** Características de la edición de Wilson. [Online] 2009. [Cited: 25 10, 2010.] <http://www.cc.uah.es/drg/adis2009/articulos/adis-09-Moreno-ISMOTE.pdf>.
- Nielsen, Tomas. 2000.** [Online] 2000. [Cited: 01 23, 2011.] http://iie.fing.edu.uy/ense/assign/recpat/material/tema3_00-01/node11.html.
- Nores, Jose Emilio Gondar. 2005.** *Redes Neuronales Artificiales*. 2005.
- Qt. 2010.** Qt. [Online] 2010. [Cited: 12 14, 2010.]
- QtCreator. 2010.** [Online] 2010. [Cited: 01 12, 2011.] <http://techerald.com/page/que-es-qt-ejemplo-clasico-hola-mundo--230320090143.html>.
- Reguera, C. 1995.** *Modelos básicos de Redes Neuronales Artificiales*. 1995.
- Rojas, Ricardo Dodds. 2009.** *Redes Neuronales. Estructura*. [pdf] Santiago de Chile : s.n., 2009.
- Salas, Rodrigo. 2007.** [Online] 2007. [Cited: 11 2010, 13.] <http://es.scribd.com/doc/36772397/Redes-Neuronales-Artificiales>.
- Slideshare. 2008.** [Online] 2008. [Cited: 01 15, 2011.] <http://www.slideshare.net/ARMANDO1022/red-neuronal-artificial>.
- Soria, Emilio. 2001.** *Redes Neuronales Artificiales*. 2001.
- Tabar, Rolando Gómez. 2005.** Slideshare. [Online] 2005. [Cited: 12 2010, 12.] <http://www.slideshare.com>.
- Valdovinos, Rosa Maria. 2006.** *Edicion de Wilson*. España : s.n., 2006.
- Vázquez, Fernando. 2008.** *Algoritmos de Aprendizaje Continuo Mediante Selección de Prototipos para Clasificadores Basados en Distancias*. España : s.n., 2008.

GLOSARIO DE TÉRMINOS

Algoritmo de entrenamiento: Proceso que le permite a una red neuronal artificial aprender de un conjunto de patrones.

Aprendizaje: Proceso mediante el cual un individuo adquiere conocimiento y experiencia, de forma que le permita modificar su comportamiento y adaptarlo a nuevas condiciones de su entorno.

Inteligencia artificial: Rama de la informática que se encarga de crear software y hardware capaces de imitar la inteligencia humana.

Red neuronal artificial: Técnica de la inteligencia artificial que simula el cerebro humano. Está compuesto por un conjunto de neuronas interconectadas que colaboran entre sí para obtener una salida.

Algoritmo de edición: Proceso que elimina los elementos atípicos presentes en un conjunto de entrenamiento.

Centroide: Punto que define el centro geométrico de un objeto.