CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE ELECTRÓNICA DEL ESTADO SÓLIDO

Análisis con Método Metaheurístico de Eventos Complejos

TESIS

Que presenta:

M. en C. Álvaro Anzueto Ríos

Para obtener el grado de
DOCTOR EN CIENCIAS
EN LA ESPECIALIDAD DE INGENIERÍA ELÉCTRICA

Directores de la Tesis:

Dr. Felipe Gómez Castañeda

Dr. José Antonio Moreno Cadenas

Ciudad de México **Diciembre del 2021**

Agradecimientos

A mis asesores:

- **Dr. Felipe Gómez Castañeda**, por su paciencia, por su manera tan peculiar y divertida de darme sus observaciones y sobre todo por su enseñanza constante. “Muchas gracias doctor”.
- **Dr. José Antonio Moreno Cadenas**, por su disponibilidad, sus observaciones y la platica constructiva que nunca falto. “Gracias doctor Moreno”.

A mis sinodales:

- **Dr. Ramón Peña Sierra**, sus observaciones enriquecieron el estado final del trabajo escrito. Gracias doctor.
- **Dr. Gabriel Romero Paredes Rubio**, las observaciones vertidas en el trabajo escrito fueron muy acertadas y constructivas. Gracias doctor.
- **Dr. Mario Alfredo Reyes Barranca**, la manera tan clara de expresar sus observaciones y los comentarios pertinentes han ayudado a mejorar el trabajo escrito. Gracias doctor.
- **Dr. Víctor Hugo Ponce Ponce**, por sus comentarios, observaciones vertidas sobre el trabajo escrito. Gracias doctor por la disponibilidad mostrada y por estar siempre de muy buen ánimo.

A las instituciones:

- Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo otorgado para la realización de este trabajo.
- Agradezco al **CINVESTAV - IPN**, por las facilidades otorgadas durante la realización de este trabajo y por la educación recibida.

Índice General

Índice de Figuras

Índice de Tablas

Resumen

0.1. Resumen

En este trabajo se contempla la reducción del número de dimensiones o características de una base de datos. Para realizar el proceso de reducción de dimensiones, se ha utilizado el algoritmo t-SNE (t distributed Stochastic Neighbor Embedding Algorithmt). La utilización consta de anexar una arquitectura con neuronas tipo Spiking con respuesta de tipo no lineal. Esta arquitectura neuronal se encarga del procesamiento de los datos y extrae de ellos las características relevantes. Las neuronas spiking empleadas en la arquitectura neuronal, son una innovación ya que presentan una respuesta de salida de tipo sigmoide y tienen como base el modelo matemático propuesto por Izhikevich.

Para el proceso de entrenamiento de la red neuronal, que consiste en determinar los valores de conductancia sináptica para cada neurona, se ha implementado el algoritmo metaheurístico de colonia artificial de abejas. Los valores de conductancia sináptica son actualizados, por el algoritmo de abejas, hasta determinar el juego de parámetros que optimizan la función objetivo empleada para el entrenamiento de la red neuronal. En este trabajo se demuestra, a través de experimentos numéricos, la respuesta superior que presentan las neuronas tipo spiking con respuesta no-lineal, al procesar datos provenientes de problemas complejos.

Con la finalidad de determinar la calidad de la distribución de datos generada por la arquitectura neuronal propuesta, se propone una comparación contra una distribución de referencia. Para cada dato, en ambas distribuciones, se comparan los valores de las características para obtener el valor de similitud; el valor de similitud global es obtenido calculando el error cuadrático medio (ECM). Para demostrar el hecho de la reducción pertinente de las dimensiones, se ha aplicado la métrica del cálculo de la matriz de co-Ranking. Finalmente, y con base en los resultados obtenidos, se demuestra que el sistema propuesto es eficiente en la tarea de procesar

y analizar grandes cantidades de datos y lograr extraer de ellos información relevante para la toma de decisiones.

0.2. Abstract

In this work, reducing the number of dimensions or characteristics in a database is developed. To carry out this task, the t-distributed Stochastic Neighbor Embedding Algorithms or t-SNE has been implemented. The implementation consists of annexing an architecture with Spiking neurons with a non-linear response. This neural architecture is in charge of data processing and extracts the relevant characteristics from it. The spiking neurons used in the neural architecture are an innovation since they present a sigmoid-type output response and are based on the mathematical model proposed by Izhikevich.

For the neural network training process, which consists of determining the synaptic conductance values for each neuron, the metaheuristic algorithm of an artificial bee colony or ABC has been implemented. The ABC algorithm updates the synaptic conductance values until the set of parameters that optimize the objective function used to train the neural network is determined. In this work, the superior response of spiking neurons with a non-linear response is demonstrated through numerical experiments when processing data from complex problems.

In order to determine the quality of the data distribution generated by the proposed neural architecture, a comparison is proposed against a reference distribution. For each data, in both distributions, the values of the characteristics are compared to obtain the similarity value; the global similarity value is obtained by calculating the root mean square error (MSE). To demonstrate the pertinent reduction of the dimensions, the metric of the calculation of the co-Ranking matrix has been applied. Finally, and based on the results obtained, it is shown that the proposed system is efficient in processing and analyzing large amounts of data and manages to extract relevant information from them for decision-making.

Objetivos

0.3. Objetivo General

Se propone Establecer una metodología de reducción de dimensionalidad de datos de eventos complejos con base en un esquema de optimización metaheurística.

0.4. Objetivos Particulares

- Desarrollar una estrategia metaheurística para el análisis de datos de eventos complejos a través de la visualización de sus componentes principales en una dimensión reducida.
- Determinar los valores de los parámetros de una neurona de Izhikevich que aproximen un nuevo comportamiento de tipo sigmoidal.
- Entrenar de forma óptima una red neuronal con neuronas de Izhikevich en la tarea de reducción de dimensión en datos de eventos complejos usando una función informática de sistemas de aprendizaje en máquina.

1 Introducción General

En los últimos años los avances tecnológicos han permitido aprovechar de mejor manera los recursos naturales. Se han optimizado las formas de consumo humano y hasta el estilo de vida en las ciudades.

En el campo de la electrónica se han investigado y propuesto nuevas técnicas que han permitido mejorar la forma de medición de parámetros en diferentes áreas. Se han investigado e implementado nuevos materiales en los sensores, extendiendo las áreas de aplicación. Un ejemplo notorio de esto, es la generación de un nuevo paradigma en la tecnología conocido como *Internet de las Cosas ó IoT (del inglés: Internet of Things).*

Al conectar distintos dispositivos a una misma red y con cada uno de ellos tomando registros de los diferentes parámetros en un ambiente, durante largas jornadas en el día, la cantidad de datos recolectada llega a ser enorme, dando pie a otro paradigma nombrado *Big-Data o almacenamiento de grandes cantidades de información.* El término de *Big-Data*, hace referencia al almacenamiento e interpretación de grandes cantidades de información. Debido al proceso de análisis e interpretación de una enorme cantidad de datos para la extracción de características o patrones, se han generado nuevas técnicas en el procesamiento de los datos, que permiten trabajar de forma paralela buscando con ello minimizar los tiempos de procesamiento.

Buscando técnicas para el procesamiento de grandes cantidades de datos se han explorado nuevas alternativas basadas en el comportamiento de algunas especies de insectos, como son abejas, hormigas y libélulas, entre otros. Se ha modelado matemáticamente su comportamiento organizacional, permitiendo con esto la generación de nuevos algoritmos con la capacidad de procesar datos de una manera estructurada. Este tipo de algoritmos se basan en el concepto de la metaheurística, donde cada individuo logra obtener una solución, para luego analizar las soluciones encontradas por un conjunto de individuos, eligiendo la mejor solución del grupo

basándose en una función que describa el resultado esperado; esta función a menudo es llamada *función objetivo ó de optmización.*

Dentro de los datos almacenados, una problemática común a la hora de determinar patrones de comportamiento, es la existencia de información redundante y que sin tener conocimiento de esto debe ser procesada, agregando con ello el consumo de recursos computacionales, por lo tanto, se vuelve indispensable la extracción de características esenciales de las que no lo son.

Para la extracción de características esenciales, un método ampliamente estudiado es el *Análisis de Componentes Principales.* Si bien para esta técnica, sus bases matemáticas se encuentran en el álgebra lineal, se han propuesto arquitecturas basadas en redes neuronales artificiales que desarrollan el mismo concepto, esto es, la extracción de las componentes principales. Una arquitectura frecuentemente empleada para este propósito es la denominada de tipo *Autoencoder.* Esta estructura neuronal se basa en tres capas neuronales. La primera, asociada a la entrada es considerada como la capa encargada de realizar la codificación de los datos, es decir, los datos son procesados y ordenados basándose en alguna característica en común, los datos son organizados bajo una secuencia establecida. La capa de salida, también conocida como capa de decodificación, se encarga de reorganizar los datos conforme a su estructura inicial; los datos a la salida son organizados con la misma secuencia con la que se presentaron inicialmente a la entrada de la arquitectura neuronal. La capa intermedia o capa profunda, se considera como una región donde se traslapa la codificación y la decodificación. Esta capa intermedia, otorga una representación compacta de los datos; se puede decir que ella contiene una secuencia comprimida o de componentes esenciales, con la que se pueden reestructurar los datos a su condición inicial. En términos simples, se extraen las componentes principales de la información. La capa intermedia contiene una versión reducida de las características que constituyen a los datos iniciales; este proceso de reducción es comúnmente conocido, en términos del análisis de datos, como reducción de dimensionalidad en la representación de los datos.

La reducción en la dimensión de las características, para la representación o descripción de los datos, debe ser vista como el proceso de extraer las rasgos o características esenciales con la que son representados los datos. A manera de ejemplo, podemos mencionar que, si un grupo de datos es representado por diez características o rasgos que los definen, la reducción de dimensionalidad se encarga de analizar cada característica para poder determinar su contribución en la descripción. Aquellas características que contribuyan en menor grados son desechadas, logrando con esto, que con una menor cantidad de información se mantenga la misma distribución

en los datos. Una menor cantidad de características, hablando en términos del procesado de los datos, representa una disminución en las operaciones matemáticas y por ende menor tiempo de cómputo. Una técnica emergente, en la tarea de la reducción de dimensionalidad, es la llamada *t-SNE* por sus siglas del inglés *t-Distributed Stochastic Neighbor Embedding*. Esta técnica se basa en representar los datos en su forma inicial, como una distribución de probabilidad de tipo T-Student (distribución de Cauchy). Propone para la salida, una reducción en la dimensión de las características; la propuesta de salida, vuelve a ser representada como una función de distribución de probabilidad y las dos distribuciones son comparadas y el valor de semejanza es quien determina si la propuesta de los datos en la salida son una versión reducida de los datos iniciales. Para el cálculo del valor de semejanza, se desarrolla la técnica numérica conocida como divergencia de Kullback-Leibler. Cuando el valor de semejanza tiende a cero, indica que las distribuciones son equiparables y por tanto los datos han conservado su distribución inicial, pero con menos datos numéricos en su representación. Si para una primera comparación, el valor de divergencia se considera alto, la distribución a la salida mantiene la cantidad de rasgos o características y tan solo se modifica el valor de la magnitud en cada una de ellas; este proceso de la modificación se realiza empleando la técnica de gradiente descendente. En este trabajo de tesis se propone utilizar la función de pérdida del tipo Kullback-Leibler como la función objetivo a ser minimizada en un proceso de entrenamiento metaheurístico de una red neuronal pulsante con neuronas de Izhikevich.

La técnica *t-SNE*, es empleada exclusivamente para la visualización de los datos en un espacio dimensional reducido y lograr determinar una distribución de probabilidad con los datos procesados.

Las redes neuronales artificiales tipo spiking, también llamadas de tercera generación, son las que mayormente aproximan el comportamiento de las redes neuronales biológicas. Estas redes neuronales ocupan modelos matemáticos basados en ecuaciones diferenciales ordinarias, para reproducir los trenes de pulsos (o spikes) de las neuronas biológicas, los cuales son identificados como los mecanismos temporales de comunicación entre ellas.

Los primeros modelos neuronales de tipo spiking, fueron basados en el comportamiento de la membrana celular que genera pequeñas variaciones de potencial al recibir una señal externa, este modelo se conoce como *Spike Response Model (SRM)*. El modelo SRM, es un modelo simplificado del propuesto por Hodgkin and Huxley. Este modelo, puede ser considerado poco plausible, ya que, al rebasar un valor de potencial de umbral, previamente establecido, pasa inmediatamente

a un estado de potencial de reposo, sin llegar a producir el spike esperado.

Un modelo mucho más aproximado al comportamiento biológico y computacionalmente menos complejo, es el propuesto por el matemático Eugene Izhikevich. Este modelo reproduce los spikes de las neuronas corticales y combina la plausibilidad biológica del modelo de Hodgking-Huxley y la velocidad de cómputo del modelo SRM, lo que lo hace atractivo no tan solo en este trabajo, sino en todos los que lo han utilizado desde su aparición. El modelo modificado de una sola neurona de tipo Spiking puede resolver problemas de tipo no lineal y aquí ha sido empleado para generar de manera sucesiva diferentes distribuciones de datos.

El trabajo de tesis presentado, tiene la premisa de proponer un método para la reducción en la dimensionalidad de los datos, basado en una arquitectura neuronal de tipo spiking permitiendo manejar bases de datos con cantidades considerables de información.

1.1. Antecedentes Históricos

El procesamiento de información ha sido una tarea en constante desarrollo y atractiva para los investigadores desde hace mucho tiempo; con el procesamiento y análisis de la información se pueden tomar decisiones o acciones para la solución de un problema. Como primer paso, es necesario tener conocimiento previo de un evento o las razones que lo generan; en el ámbito de la ingeniería esta tarea se desarrolla con la recopilación de datos. Los datos son representaciones numéricas de rasgos o características que definen un evento, por lo tanto, pueden ser representados de forma gráfica, facilitando su interpretación y análisis. Los valores numéricos pueden ser procesados en sistemas computacionales, que mediante algoritmos extraen comportamientos o patrones de los eventos.

El valor numérico de un dato, va más allá de tan solo pertenecer o no pertenecer a un evento. Un dato o conjunto de datos, puede definir el comportamiento de más de un evento o acción, por lo tanto, se considera que tienen diferentes niveles de influencia en la toma o ejecución de una posible acción.

En 1965, Lotfi Asket Zadeh, introduce la teoría de *"Lógica Difusa"* [1] en la cual se presenta una forma extendida del análisis de datos, considerando el esquema de valores multivaluados o de múltiple influencia en la representación de un evento. Esta teoría fue muy aceptada por la comunidad científica y fue aplicada en muchos campos de la ingeniería. El análisis, no solo del valor numérico, sino de su interpretación dio pie al avance tecnológico de otras actividades, como

la comercial [2], la construcción [3], transporte colectivo de personas [4], mejora del proceso de las comunicaciones [5], entre muchas otras. En la actividad comercial se recopilaron datos en centros comerciales que indicaban la forma del consumo de múltiples productos y con ellos se generaban campañas comerciales. En el campo de la construcción se analizaron nuevos materiales en diferentes ambientes naturales; con la captura de los datos se proyectaron nuevas formas de construcción y elección de los materiales dependiendo del medio. En la ciudad de Sendai, Japón, el tren urbano nombrado *"Sendai Subway 1000N Series"*, se construyó con un sistema de control de velocidad basado en lógica difusa y para ello se requirió el análisis de datos sobre los materiales de fabricación de los rieles, vibraciones y parámetros asociados al desarrollo del control de velocidad. Estas aplicaciones dieron lugar el desarrollo de nuevo algoritmos para el procesamiento de los datos, que en esencia buscan características y patrones de comportamiento en ellos.

Como idea de partida se analiza el *"agrupamiento de datos"*, esta tarea se lleva a cabo considerando que un grupo de datos comparte algún rasgo o característica en común, lo cual los convierte como grupo que define una clase o un tipo de evento con características definidas; si más datos son analizados y las condiciones son invariantes se puede llegar a predecir la ocurrencia de eventos. De los primeros algoritmos propuestos para el agrupamiento de datos, podemos mencionar el desarrollado por James C. Bezdeck [6], nombrado *"Fuzzy C-Means"*, el cual analiza por medio de la lógica difusa el agrupamiento de los datos, con el inconveniente de que es necesario, como parámetro inicial, definir el número de clases o grupos en que se organizarán; esto limita a que se debe conocer de manera previa la distribución de los datos y no tiene la capacidad de generar de forma dinámica nuevos grupos. Tratando de dar solución a este problema, Bezdeck en un segundo trabajo plantea el concepto de *"Cluster Validity"* [7], en el cual se calcula la distancia entre los centros de las clases y lo compacto de cada una de ella; si los datos, dentro de una clase formada, tienen una dispersión considerable, es factible realizar una subdivisión de dicha agrupación. Este trabajo abrió la posibilidad de generar nuevos algoritmos con la capacidad de modificar el número de clases o grupos definidos de forma inicial y proponer la creación de nuevos.

El análisis y agrupamiento de datos derivó en la idea de procesar los datos de manera secuencial, es decir, un primer dato por sí solo, se considera una clase. El siguiente dato es comparado, calculando la distancia con respecto al anterior; si se encuentra dentro de una distancia previamente establecida, se realiza la consideración de que ambos pertenecen a una misma clase,

de lo contrario se tendrán dos clases, este planteamiento se conoce con el término de "Cuantización vectorial" [8]; sin embargo, tiene el inconveniente de depender en gran medida del valor de distancia propuesta a considerar como perteneciente a una clase, limitando a que los grupos formados deban de ser de tamaños similares, lo cual es poco común.

El finlandés Teuvo Kohonen en [9] propone el algoritmo de aprendizaje *"Self Organizing Maps"* que desarrolla la autoorganización de los datos; este algoritmo rápidamente se convirtió en un referente sobre sistemas autoorganizados y a la fecha sigue siendo un algoritmo de comparación, se ha aplicado en las redes neuronales artificiales y las dota con la capacidad de un aprendizaje sistematizado e iterativo para el análisis de datos, evitando además el inconveniente de que las clases deban tener distribuciones similares.

Otro punto en el que se ha trabajado, es la forma de la recopilación y almacenamiento de los datos; algunos años atrás, no se contaba con una estructura establecida para esta tarea, tan solo se recopilaban los valores numéricos sin importar, en la mayoría de los casos, la forma en que eran generados. Por citar un evento, si múltiples tiendas vendían un producto, no importaba la tienda de procedencia, únicamente la cantidad de productos vendidos; hoy en día los datos recopilados tienen una estructura de almacenamiento, es decir, ahora se cuenta con la información de la tienda que ha vendido el producto, la hora y la región de mayor consumo de un producto en particular, nombrando actualmente a este tipo de recopilación como datos estructurados. La recopilación de información dio origen a bases de datos con la posibilidad de aceptar modificaciones dinámicas, es decir, pueden modificarse dentro de ellas el número de características, cantidad de datos y subdivisiones internas dependiendo de algún parámetro de almacenamiento, como puede ser el tiempo de duración de la captura, hora recurrente de una captura en diferentes días o la posición geoespacial del dispositivo que realiza la captura del dato. Esta versatilidad en las bases de datos permite analizar toda la información recabada desde diferentes perspectivas, lo que dio lugar a determinar que, no todas las característica o rasgos pueden inferir en el análisis de la información, dando lugar con ello al concepto de reducción de dimensionalidad.

Para el proceso de reducción en la dimensionalidad se propusieron originalmente algoritmos basados en los análisis estadísticos de los datos. De esta manera apareció la técnica nombrada como *"Análisis de Componentes Principales"* [10], la cual desarrolla un proceso matemático obteniendo del conjunto de datos primeramente la matriz de covarianza, para luego extraer de esta matriz los valores y vectores propios del conjunto de datos. Los vectores y valores propios obtenidos indican los ejes o dimensiones, en los cuales se tiene una mayor variabilidad,

lo que indica que sobre ese eje o dimensión los datos se separan, pudiendo de esta manera reconocer sobre qué ejes los datos pueden presentar separación en las posibles agrupaciones, en caso contrario, un eje con poca variabilidad nos indica que los datos se encuentran aglomerados, representando que, bajo la consideración de esa característica o rasgo, los datos no presentan diferencia, por lo tanto puede ser descartada en el análisis global de la información.

Las técnicas mencionadas hasta este punto, dieron soluciones satisfactorias para bases de datos compactas, que no superaban los kilobytes de almacenamiento y con una baja cantidad de rasgos o características en la representación de los datos. Con la llegada del internet, se facilitó el proceso de compartir datos, lo que propició un incremento sustancial en los volúmenes de información y la necesidad de menores tiempos para su procesado, revolucionando de esta manera el manejo y tratamiento de los datos. Con lo descrito podemos decir que, sigue vigente la motivación por generar algoritmos que aumentan la capacidad para el procesado de los datos, realizar técnicas que ayuden a disminuir las dimensiones en su representación y consumir menos recursos computacionales, lo cual representa menor tiempo de ejecución de los algoritmos; una muestra de ello es la propuesta presentada en este trabajo.

1.2. Estado Actual del Conocimiento

Como ya se ha comentado, el procesamiento, análisis e interpretación de los datos es una tarea de mucho interés para la comunidad científica y, por lo tanto, es un tema vigente abordado desde diferentes perspectivas. En este apartado se realiza una descripción de los trabajos que se han revisado de los últimos años, para tener un contexto actual del tema.

Considerando a las redes neuronales, en [11–15] presentan avances en las redes neuronales para el análisis de grandes cantidades de información, aún con los nuevos esquemas se mantienen los problemas principales: altos tiempos de procesamiento y el consumo de recursos computacionales. Una sola unidad neuronal artificial convencional, aunque pueda ser diseñada en esquemas de trabajo paralelo, únicamente puede dar una respuesta, por lo que para grandes cantidades de datos se generan redes neuronales que contemplan un número elevado de neuronas para su configuración.

Una neurona spiking, tiene la capacidad de generar pulsos como respuesta ante la presencia de una excitación o valor numérico en su entrada; si la excitación cambia, se puede modificar la frecuencia de los pulsos y, en algunos casos, la forma de los mismos, por lo que una sola neurona

logra tener diferentes respuestas para diferentes entradas. Para el procesado de datos se logra reducir la cantidad de neuronas a emplear en comparación con las convencionales. Las neuronas artificiales de tipo spiking, también conocidas como de tercera generación, son plausiblemente más apegadas al comportamiento de las de tipo biológicas. El matemático Eugene M. Ishikevich en [16] plantea un modelo matemático que logra asemejar la forma de los pulsos y las frecuencias de las neuronas biológicas. Tomando como base lo descrito en el modelo, se propone un juego de valores numéricos en los parámetros que hacen que el modelo tenga una respuesta tipo sigmoide.

Carlson y Carin en [17], definen el término de "Spike Sorting" como el proceso de extraer trenes de picos neuronales a partir de datos electrofisiológicos registrados por electrodos implantados en el tejido cerebral extracelular y plantean la relación del comportamiento en la generación de pulsos, al trabajar con grandes cantidades de información, planteando con esto el uso de sistemas spiking en el procesamiento de datos.

Actualmente con el crecimiento de las tecnologías y el internet, se ha desarrollado el paradigma del Internet de las Cosas, y se ha aplicado en diferentes contextos. Un ejemplo de ello lo encontramos en el manejo de los recursos en las ciudades. Vodak y sus colaboradores, en su trabajo [18] describen los avances tecnológicos que han propiciado el uso de sistemas basado en el Internet de las Cosas o IoT (del término inglés: Internet of Things), en las llamadas ciudades inteligentes. En la agricultura, en donde se han introducido el monitoreo de los parámetros en la calidad de la tierra y el proceso en el crecimiento del cultivo [19], se ha demostrado que IoT favorece a una mayor producción y reducción en los costos. El monitoreo constante y la necesidad de almacenar toda la información generada da paso a la creación de bases de datos cada vez de mayores envergaduras, sin olvidar mejorar los sistemas de transferencias de datos.

Steven Oberlin en [20], considera que para aplicaciones en Big-Data es factible implementar algoritmos avanzados de aprendizaje en máquina (de la terminología inglesa: "Machine Learning"), aprovechando la capacidad de este tipo de algoritmo de incluir búsqueda de relaciones entre datos, la aplicación de base de conocimiento, y la capacidad de incluir procesos matemáticos que ayudan a minimizar los efectos de la presencia de ruido en los datos. El trabajo aquí presentado parte de una idea similar, ya que incluye una metodología que tiene la capacidad de reconocer la redundancia en datos y determinar qué características son esenciales en la toma de decisiones y cuáles no.

Hinton y Rowies en 2002, presentan el trabajo titulado "Stochastic Neighbor Embedding", [21], en el cual, consideran un enfoque probabilístico para la tarea de agrupar objetos o eventos,

que son descritos por vectores de alta dimensionalidad considerando el valor de disimilitud entre ellos, para luego poder representarlos en espacios de baja dimensión de manera que preserven su identidad y la de sus vectores vecinos. El concepto de disimilitud es empleado para minimizar el efecto de los vectores con ruido, ya que si un vector tiene un valor alto para todas las clases es desechado. Consideran funciones de probabilidad gaussiana para realizar su proceso. Cada vector es considerado como el centro de una función gaussiana y se calcula la probabilidad del resto de los vectores a la pertenencia del vector inicial, pero esta consideración provoca que los vectores más cercanos contribuyan mayormente en la totalidad de valor de probabilidad y que los vectores lejanos su contribución sea casi nula, lo cual afecta al analizar la disimilitud entre un vector y las clases formadas. En un segundo trabajo Hinton y Van Der Maaten [22], realizan una extensión y proponen utilizar una distribución de probabilidad "t-studen" (distribución de probabilidad de Cauchy), la cual mejora el análisis de la disimilitud entre vectores y clases. La idea principal para la reducción de dimensionalidad, en ambos trabajos, es generar una segunda distribución de datos considerando una representación con menos dimensiones.

Al presentar una segunda distribución en dimensiones reducidas, surge la pregunta de cómo evaluar cuantitativamente que el proceso de reducción es adecuado. Lueks y colaboradores en [23] emplean el cálculo de la matriz de Co-Ranking para evaluar la pertinencia en la reducción de la dimensionalidad de los datos. El cálculo de matriz de Co-Ranking es actualmente empleada para evaluar sistemas que desarrollan la reducción de dimensionalidad [24], por lo que es considerada en este trabajo para evaluar las reducciones obtenidas.

Con lo descrito en esta sección se puede afirmar que el trabajo propuesto cubre diferentes perspectivas en el ámbito actual del procesamiento, análisis e interpretación de datos de alta dimensionalidad, lo cual indica que éste se encuentra en la frontera del conocimiento en el área de desarrollo de sistemas de procesamiento inteligente de datos en eventos complejos.

1.3. Eventos Complejos

La mayoría de los sistemas electrónicos y computacionales están diseñados bajo la ideología de Orden-Acción-Respuesta, un sistema (también nombrado método en sistemas informáticos) hace el llamado a otro sistema y le otorga instrucciones para generar una acción, produciendo de esta manera un *"evento"* o respuesta programada. Esta interacción puede ser descrita y conocida en arquitecturas computacionales; sin embargo, para el mundo real esto funciona de una manera

completamente distinta. Para el mundo real los *"eventos"* pueden darse de manera aleatoria y, para la mayoría de las veces, sin conocer de manera exacta las reacciones que estos pueden provocar. Además, se pueden clasificar como eventos temporales o fijos. Los eventos temporales están determinados por el momento de ejecución y la duración de ellos, recabando de esta manera datos que proporcionan información valiosa del evento. Esta información también se interpreta como una o múltiples observaciones del evento, sin olvidar que un evento puede desencadenar un conjunto de nuevos eventos. Por otra parte, un evento fijo se puede definir como una orden asociada a una respuesta simple; para sistemas de cómputo se puede considerar que un evento simple solo genera un dato de información, el tiempo en que se ejecutó. Como ejemplo, podemos citar el viaje de una persona en su automóvil, al ejecutar la acción de colocar la llave en el sistema de ignición se da el evento de encender el vehículo; la computadora a bordo del vehículo puede trazar la ruta del viaje deseado, desencadenando todo un conjunto de eventos, como puede ser almacenar la posición geoterrestre del vehículo, calcular el tiempo de viaje, además de poder conectarse con otros sistemas, que son ejecutados como parte de otros eventos, y otorgar información de las calles o avenidas con poco tráfico. El evento inicial de encender el vehículo ha desencadenado un conjunto de acciones o procesos que pueden ser controlados por el usuario, como lo es mover el vehículo; es importante resaltar que existirán otros eventos que salen del control del usuario, pero influirán en su toma de decisión, como lo es determinar qué calles presentan un menor tráfico, convirtiendo estos últimos eventos en los nombrados *"eventos complejos"*. Con lo descrito, podemos determinar que un *"evento complejo"* es un evento que está asociado a la ejecución o a la interpretación de otros eventos que no conservan una relación directa, sin embargo, influyen en su acción (o toma de decisión) determinando de esta manera su respuesta. Podemos determinar 5 propiedades que son asociadas a los eventos complejos.

- Los eventos complejos no son aislados, están acompañados por uno o varios eventos que se ejecutan en diferentes tiempos, pero aún con esto, se coordinan para su análisis e interpretación.

- En los eventos complejos importa el contexto en el que son ejecutados. Cuando son ejecutados es importante determinar que el evento funcione, que dé la respuesta esperada y sobre todo entender lo que está ocurriendo. Un ejemplo de ellos es interpretar la medición de un sensor que mide el nivel de suciedad en el agua, para una lavadora de ropa, y determinar la cantidad de detergente requerido.

- Para un evento complejo, es necesario organizar los datos generados de una manera estructurada y organizada, ya que es importante determinar el momento de la generación de los datos y a menudo las secuencias de ocurrencia.

- Un evento complejo requiere de una vigilancia contante, ya que los datos pueden variar sin una directriz y generar nuevas tendencias y actividades en cualquier momento.

- En relación al punto anterior, los eventos complejos conllevan a desarrollar contantemente nuevos paradigmas tecnológicos, haciendo alusión a la frase "renovar o perecer".

En este trabajo se han considerado estas 5 propiedades de los eventos complejos, lo cual nos ha llevado a todo un conjunto de innovaciones. Como primer punto, se han recopilado diferentes bases de datos, considerando diferentes escenarios de prueba. El segundo punto considerado, es la propuesta de una nueva configuración en el modelo matemático de una red neuronal de tipo Spiking perteneciente a una arquitectura neuronal que procese las bases de datos recopiladas. Finalmente, se ha presentado una nueva estructura computacional que desarrolla la tarea de la reducción de dimensionalidad en las características o rasgos que describen los datos generados por eventos complejos. Con las propuestas planteadas en este trabajo, se da el cumplimiento a la idea de una constante innovación.

1.4. Conclusiones del Capítulo

En este capítulo se han abordado de manera histórica, las principales técnicas para el procesamiento de datos y como se ha modificado, para que además se puedan extraer de ellos patrones de comportamiento que ayudan en la toma de decisiones.

Se ha descrito la importancia que llegan a tener los datos cuando son generados por diferentes fuentes o eventos pero que, en sumatoria, llegan a influir en eventos de mayor complejidad.

Finalmente, se puede determinar que con el desarrollo tecnológico actual, se ha incrementado geométricamente el almacenamiento de información y esto ha propiciado que surjan nuevas estrategias para el procesamiento y análisis de datos, sin embargo, las técnicas precedentes han cimentado las bases de la extracción de características en las bases de datos y por ello, el actual capítulo ofrece un panorama para innovar en el campo del aprendizaje y tratamiento de información de manera autónoma, teniendo como una de sus principales directivas el poder procesar grandes cantidades de información.

2

Optimización Metaheurística

La palabra heurística proviene del griego, etimología que comparte con eureka, que significa, "encontrar, hallar, descubrir, comprender, inventar, etc". En una definición simplificada podemos mencionar "el arte de inventar e innovar".

En este capítulo, se describe la implementación de algoritmos metaheurísticos basados en el comportamiento grupal que se presenta en algunos seres en la naturaleza, para ser aplicados en la solución de problemas de optimización. En este capítulo se dan tres ejemplos de aplicaciones en la ingeniería.

2.1. Definición de Metaheurística

Partiendo del concepto de heurística relativo a "Idear, Imaginar, Crear, Inventar", tenemos el concepto de su aplicación como heuresis, "Solucionar, Visualizar, Generar, Innovar", sin embargo, una definición más apegada a la ingeniería sería como una colección de técnicas, procedimientos matemáticos o métodos secuenciados, para resolver un problema dado.

Para el desarrollo de algoritmos o procedimientos secuenciados es común considerar reglas heurísticas, donde se consideran sugerencias o advertencias según conocimientos previos. Un ejemplo del uso de la heurística se puede encontrar en el paradigma de Lógica Difusa, en el cual se construye una base o memoria de conocimiento, que básicamente es la unión de reglas o restricciones basándose en el conocimiento previo y la experiencia que se tiene sobre un problema o tarea que no se encuentra claramente definida. Apegado al proceso que realizan algunos seres vivos en la naturaleza, la heurística se considera como la experiencia que desarrolla cada ente y la experiencia de otros entes, del mismo grupo, para determinar la solución que se presente en su ambiente, es decir, se comparte el conocimiento.

Por lo anterior podemos decir que la heurística es una técnica, método o procedimiento para resolver una tarea o problema, el cual no se encuentra claramente definido, y que no es producto de un análisis formal, sino de conocimiento experto sobre el tema.

Metaheurística: El nombre combina el prefijo griego “Meta” (“más allá”, “a un nivel superior”, “por encima de”) y "heurístico". Por lo tanto, en una definición simplificada, podemos decir que la metaheurística consiste en estrategias generales para construir algoritmos, que ofrezcan mejores resultados o que tengan un mayor desempeño respecto a alguna métrica, que las estrategias heurísticas.

Una idea genérica de la metaheurística está relacionada al concepto de resolver una tarea o problema, empleando conocimiento y/o agregando conocimiento proveniente de nuevas fuentes, mejorando de esta forma los resultados obtenidos al considerar una sola fuente. En términos de algoritmos matemáticos podemos definir a la metaheurística como un conjunto de estrategias generales para la solución óptima de problemas, teniéndose un alto grado de rendimiento.

Las estrategias metaheurísticas pueden se clasificadas en tres ramas:

- Metaheuristicas Basadas en Métodos Constructivos.
- Metaheuristicas Basadas en Trayectorias.
- Metaheuristicas Basadas en Poblaciones.

Los algoritmos autoorganizados pueden ser considerados como un ejemplo de métodos constructivos, ya que se parte sin conocer la solución y por cada evento ejecutado, se recopilan y se analizan los resultados, para construir una solución. Un ejemplo de métodos constructivos lo tenemos en los algoritmos basados en colonia de hormigas (Ants Colony Optimization) [25] y GRASP (Greedy Randomized Adaptive Search Procedure) [26].

La metaheurística basada en trayectorias se refiere a un algoritmo de búsqueda de secuencias locales; se considera tender un mallado en el espacio de búsqueda y se determina la secuencia de nodos para generar una ruta de solución. Para estos sistemas, se considera siempre un punto de partida y un punto final; el punto de partida generalmente está establecido, pero el punto final puede variar, así como también, la ruta para llegar a él. Durante el proceso de búsqueda de solución se generan múltiples rutas o secuencias y para determinar la calidad de una ruta propuesta, es necesario contar con una métrica. Ejemplos de este tipo lo encontramos en los algoritmos de búsqueda TABU (Tabu Search) [27], algoritmo de búsqueda de alimento de hormigas (Ants Colony Optimization).

En la metaheurística basada en poblaciones se consideran múltiples agentes buscando el punto de solución; un agente proporciona una solución al sistema y, al final de un ciclo de ejecución, todas las soluciones son comparadas y mediante una métrica se determina la mejor de todas. Este tipo de metaheurística puede ser implementada de forma paralela ayudando con esto a minimizar los tiempos de ejecución. Ejemplos de este tipo de algoritmos lo tenemos en los algoritmos de agrupación de enjambre de partículas (Particle Swarm Optimization) [28] o por colonia artificial de abejas (Artificial Bee Colony) [29], entre otros.

Los algoritmos metaheurísticos combinan, de forma apropiada, distintos conceptos para explorar y explotar eficientemente el espacio de búsqueda. Para estos algoritmos podemos citar algunas de sus ventajas e inconvenientes.

Ventajas:

- Son algoritmos no específicos y adaptables a diferentes espacios de trabajo.
- Fácil implementación.
- Sistemas paralelos.
- Eficientes para resolver problemas complejos.
- Manejo eficiente de problemas con presencia de gran cantidad de datos.

Inconvenientes:

- Son algoritmos aproximados, no exactos.
- Son no determinísticos (probabilísticos).
- No siempre existe una base teórica establecida, por lo que muchos de ellos son desarrollados a partir de observaciones de entes en la naturaleza.

Los sistemas metaheurísticos, se apoyan de las estrategias para la búsqueda de soluciones de tipo explotación y exploración, y lograr combinarlas para, de forma iterativa, aproximarse a la obtención de la solución de un problema. La estrategia de búsqueda de solución de tipo de explotación, también nombrada como intensificación, se describe como búsqueda en la región actual y su vecindario cercano. Este concepto se ve representado en la Figura 2.1.

La estrategia de búsqueda de solución de tipo exploración, también nombrada como diversificación, puede definirse como la técnica de explorar regiones distantes del espacio de solución (Ver Figura 2.1).

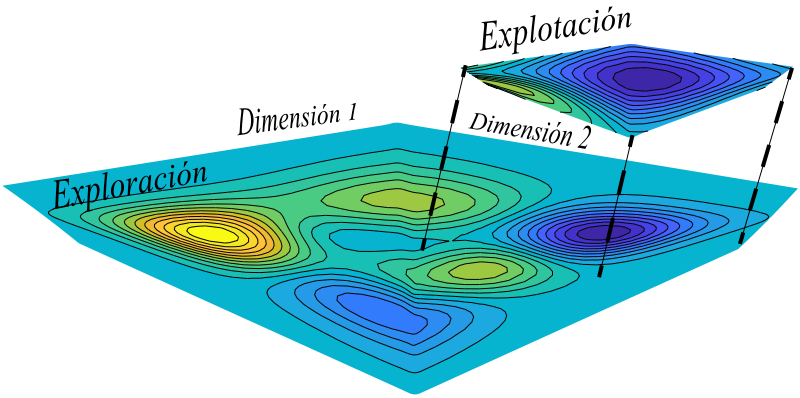

Figura 2.1: Exploración y Explotación de la Región de Interés

De forma simplificada, podemos asociar el término de explotación como búsqueda local y el término de exploración como búsqueda global.

2.2. Organización por Colonia de Hormigas (ACO-R)

2.2.1. Introducción de ACO-R

El análisis de grandes cantidades de información, a menudo es nombrado como "Minería de datos" y hace alusión a los procesos exhaustivos de registrar y revisar datos y las posibles relaciones entre ellos. La minería de datos puede ser abordado por Metaheurística, la cual ha demostrado una tendencia a resolver de forma eficiente las tareas de cómputo propias solicitadas.

El grupo de Sistemas VLSI de la SEES, ha desarrollado en forma de plataformas digitales con tecnología FPGA, algunos prototipos para el reconocimiento de patrones pertenecientes a sistemas de visión y otros similares. En ellos se han utilizado modelos de redes neuronales artificiales y de lógica difusa. También, se han incluido temas de uso de algoritmos metaheurísticos, los cuales optimizan la respuesta de sistemas de clasificación y el análisis de datos.

En la naturaleza, las hormigas exploran su entorno en busca de fuentes de alimento. Este comportamiento de búsqueda de alimento ha inspirado una variante del sistema inteligente metaheurístico llamado algoritmo de optimización por colonia de hormigas (del inglés: Ants Colony Optmization, ACO). En 1999, los investigadores Dorigo, Di Caro y Gambardella, determinaron que las hormigas se comunican unas con otras para intercambiar información sobre el estado de las fuentes de alimentos a través del depósito de feromonas [30].

La información del estado se intercambia dentro de un ámbito local. Una hormiga puede informar a su vecina del hallazgo de la fuente de alimento. Después de encontrar la comida esta

hormiga regresa al nido, dejando a su paso un rastro de feromonas para que otras hormigas puedan seguirla. Al descubrir el rastro de comida, otras hormigas vecinas abandonan su búsqueda aleatoria de comida y siguen el rastro de feromonas, reforzando de esta manera el camino establecido.

Con el tiempo, sin embargo, el rastro de feromonas tiende a evaporarse y perder su propiedad atractiva. Si el tiempo de ida y vuelta entre la colonia de hormigas y la fuente de alimento es grande, la feromona puede evaporarse más rápido de lo que puede reforzarse. En contraste, un rastro corto de feromona tiene una mayor densidad de feromona y puede reforzarse más rápido de lo que se evapora. El proceso de evaporación de feromonas es análogo a evitar la convergencia a un óptimo local. Esta propiedad es modelada para generar el algoritmo de búsqueda y optimización de rutas.

En ACO, las relaciones entre señales generadas y hormigas, están determinadas estocásticamente por medio de su regla de transición de un estado a otro, calculando la probabilidad para la toma de decisión de camino; esta elección de ruta constituye el proceso de aprendizaje hacia las regiones más nutridas en feromonas, que representa la selección de una ruta de menor coste energético para las hormigas.

2.2.2. Desarrollo de ACO-R

El paso de un estado a otro está determinado por el cálculo de la probabilidad y éste representa la toma de decisión a qué nuevo estado se desplaza una hormiga en la búsqueda de una ruta para llegar al alimento. La probabilidad de que una hormiga k pase del estado (o nodo) i al estado(nodo) j en el momento t, está definida por la Ecuación 2.1.

$$P_k\left(i,j,t\right) = \frac{\tau_{i,j}\left(t\right)^{\alpha} * d_{i,j}\left(t\right)^{\beta}}{\sum_{n\,\epsilon\,N} \tau_{i,j}\left(t\right)^{\alpha} * d_{i,j}\left(t\right)^{\beta}},\ si\ j\ \epsilon\ N_i^k \tag{2.1}$$

Donde:

- $\tau_{i,j}(t)$ = rastro de feromona entre el nodo i y el nodo j en el tiempo t
- $d_{i,j}(t)$ = distancia entre el nodo i y el nodo j
- $N_i^k(t)$ = vecindario de nodos que puede visitar la hormiga k desde el nodo i
- α = factor de peso de la feromona $(0-1)$; mientras mas cercano al valor de uno, mayor peso tendrá la feromona en la elección de una conexión
- β = factor de peso de la distancia $(0-1)$; mientras mas cercano al valor de uno, mayor peso tendrá la distancia en la elección de una conexión

Cuando ya se ha establecido la ruta para llegar el alimento, las hormigas desarrollan ahora el proceso de liberar feromonas para indicar la ruta de regreso al nido o, en un segundo caso, reforzar el camino con esta sustancia. Para modelar la cantidad de feromonas entre el nodo i y el nodo j se emplea la Ecuación 2.2.

$$\tau_{i,j}(t+1) = \rho * \tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{k,i,j}(t) \mapsto \ si\ (i,j) \in R_k : \tau_{k,i,j} = \frac{1}{L_{k,i,j}} \tag{2.2}$$

Donde:

- ρ = constante de evaporación (0-1), representa la velocidad de evaporación de la feromona
- $L_{k,i,j}(t)$ = longitud total recorrida por la hormiga k para llegar a la fuente de alimento
- $\tau_{k,i,j}(t)$ = cantidad de feromona depositada por la hormiga k entre el nodo i y el nodo j
- R_k = Ruta de la hormiga k que transita entre los nodos (i,j)
- $\Delta\tau_{k,i,j}(t)$ = Representa el incremento de feromona entre los nodos (i,j) después de una iteración

En [31], se presenta una variante del algoritmo ACO, ya que éste, de forma inicial fue pensado para resolver problemas de optimización en el espacio de dominios discretos. La variante tiene la capacidad de resolver problemas en el espacio de dominios continuos, permitiendo ser aplicado a la búsqueda de soluciones en ecuaciones de optimización. En la Tabla 2.1, se representa la matriz de solución; esta tabla contiene el número de soluciones S_k y cada solución se forma con el vector de variables o características n. $f(S_j)$ es el valor de la función objetivo que se obtiene al evaluarla con una solución, cuya ponderación o pertinencia de ese valor está definido por ω_j, lo cual nos indica la calidad de la solución propuesta [32].

Tabla 2.1: Matriz de Solución ACO-R

S_1	s_1^1	s_1^2	$\cdots$	s_1^i	$\cdots$	s_1^n	$f(S_1)$	ω_1
S_2	s_2^1	s_2^2	$\cdots$	s_2^i	$\cdots$	s_2^n	$f(S_2)$	ω_2
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
S_j	s_j^1	s_j^2	$\cdots$	s_j^i	$\cdots$	s_j^n	$f(S_j)$	ω_j
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
S_k	s_k^1	s_k^2	$\cdots$	s_k^i	$\cdots$	s_k^n	$f(S_k)$	ω_k
	g^1	g^2		g^i		g^n		

A continuación, se presenta el seudocódigo del algoritmo ACO-R1, contemplando los pa-

rámetros establecidos en la Tabla 2.1. Los resultados de la evaluación de este algoritmo son presentados en la última sección del capítulo y las líneas de código para la implementación son recopiladas en el apéndice A.

Algorithm 1 ACO-R1

ACO-R: $(k, n, m, \Delta x, f(.), \varepsilon)$ parámetros iniciales;
k : Número de soluciones,
n : Número de características,
m : Número de nuevas soluciones,
$f(.)$ **:** Función a optimizar,
ω **:** Peso de la solución propuesta,
ε **:** criterio de paro;
Begin
    for $i = 1$ to k **do**
        for $j = 1$ to n **do**
            $S_{i,j} := rand\,(min,\, max)$
        end for
        $f_i = function(S_i)$; *evaluar la función a optimizar*
    end for
    $S := sorting(S)$; *arreglar de menor* $\rightarrow$ *mayor*
    repeat
        for $l = 1$ to m **do**
            for $i = 1$ to n **do**
                elección aleatoria de una solución $S_{i,j}$, considerando $j \in [1, k]$
                $S'_{l,i}$ = modificar característica i
            end for
            $f_i = function(S'_i)$; evaluar la función a optimizar
        end for
        determinar la calidad de $S'_{l,i} vs S_{i,j}$
        $S := sorting(S)$; arreglar de $menor \rightarrow mayor$ eliminar la solución de menor valor.
    until ε, criterio de para se alcanzo
    return valor mayor del grupo de soluciones S
End

2.3. Enjambre de Partículas (PSO)

2.3.1. Introducción de PSO

En la naturaleza, muchos insectos y otros seres vivos, presentan la condición de enjambre al momento de recolectar alimento o en su consumo. Los bancos de peces presentan la condición de enjambre por dos razones principales, el consumo de alimento sistematizado y eficiente, y como medio de protección. En parvadas, la condición de enjambre es empleada para la búsqueda de alimentos y, en los viajes de migración la formación de tipo "V" que desarrollan, sirve para minimizar los efectos del viento en contra. Para el enjambre, cada agente individual o individuo

brinda la información relativa del ambiente, generando de esta manera una búsqueda o interpretación local; sin embargo, esta información es compartida con el resto de los agentes por lo que se dice que un enjambre es una colección de agentes interactuando. En su conjunto, el enjambre desarrolla una exploración global del ambiente.

En ingeniería, se acuñó el término de inteligencia artificial de enjambre (del término inglés: Swarm Intelligence) por primera vez en el trabajo de Beni y colaboradores [33]; en su trabajo, presentan el comportamiento autoorganizado de un grupo de robots. Los robots tienen capacidades limitadas de procesamiento, pero tienen programado secuencias de agrupamiento que les ayuda a realizar tareas de cooperación. La definición de inteligencia de enjambre, ellos la centran en considerar agentes poco inteligentes con capacidades de procesamiento limitadas, pero que poseen comportamientos que colectivamente son inteligentes. Una definición más completa es presentada por Kovacina en [34] y considera que la inteligencia de enjambre es: *"un grupo de agentes cuyas interacciones colectivas magnifican los efectos de los comportamientos de los agentes individuales, dando como resultado la manifestación de comportamientos a nivel de enjambre más allá de la capacidad de un pequeño subgrupo de agentes"*.

Este concepto de inteligencia de enjambre, fue retomado por los investigadores Kennedy y Eberhart [35], para proponer un algoritmo metaheurístico para la solución de problemas de optimización y lo nombraron Optimización por Enjambre de Partículas (del término inglés: Particle Swarm Optimization, PSO). Los autores en su trabajo, simulan el comportamiento social de una bandada de aves o banco de peces y desarrollan las expresiones matemáticas que describe este comportamiento, logrando sintetizarlo en un conjunto de secuencias establecidas.

2.3.2. Desarrollo de PSO

En el algoritmo de optimización por enjambre de partículas, un agente o individuo es considerado como una partícula y se desarrolla una optimización global de tipo estocástica, basándose en la simulación del comportamiento social de un enjambre. Kennedy y Eberhart [35] describen algorítmicamente el comportamiento social que presentan algunas especies de animales o insectos, cuando se encuentran agrupados en gran cantidad; la agrupación tiene comportamiento colectivo en sus movimientos logrando moverse como un solo ente y, durante su desplazamiento, las partículas demuestran coordinación evitando colisionar entre ellas y con objetos externos a la agrupación. El algoritmo de Optimización por Enjambre de Partículas consta de dos pasos principales:

- Generación inicial y de manera aleatoria de la posición y velocidad de cada partícula.
- Actualización de la velocidad, considerando el mejor resultado registrado por la partícula y el mejor resultado del grupo de partículas.

Esto es:

Primero, las posiciones generadas para las partículas estarán designadas como x_k^i y las velocidades como v_k^i; ambos parámetros se encuentran restringidos por los valores máximos y mínimos del espacio de búsqueda o solución y las ecuaciones (2.3) y (2.4) son empleadas para obtener los valores iniciales, respectivamente:

$$x_0^i = x_{min} + rand \cdot (x_{max} - x_{min}) \tag{2.3}$$

$$v_0^i = \frac{x_{min} + rand \cdot (x_{max} - x_{min})}{\Delta t} = \frac{posición}{tiempo} \tag{2.4}$$

Las posiciones y velocidades están expresadas en forma de vectores con el superíndice i que representa el número de partícula y el subíndice k que indica el número de tiempo o ciclo de ejecución, $rand$ es una variable con distribución aleatoria uniforme entre $[0-1]$. Con el proceso de inicialización descrito, las partículas pueden distribuirse sobre todo el espacio de búsqueda o solución.

Segundo, dada la posición inicial se calcula una nueva posición basándose en dos aspectos. El primero se basa en su interacción con los vecinos próximos y, el segundo, en la dirección de la agrupación, es decir, se tiene un desplazamiento influenciado de forma local y a su vez en uno global, respectivamente. Para la actualización de las velocidades se considera la ecuación (2.5).

$$v_{k+1}^i = w_p v_k^i + c_1 rand_1 (p_{mejorlocal}^i - x_k^i) + c_2 rand_2 (G_{mejorglobal} - x_k^i) \tag{2.5}$$

En la ecuación (2.5), el parámetro w_p representa el control de la inercia, c_1 y c_2 son constantes asociadas al comportamiento cognitivo y social respectivamente. Los términos $rand_1$ y $rand_2$ son valores aleatorios que están delimitados a un rango de valores entre cero y uno; con ellos, se determina la contribución del movimiento local y global que actúa sobre cada partícula. $p_{mejorlocal}^i$, determina la mejor solución registrada para la partícula i, mientras que $G_{mejorglobal}$ representa la mejor solución del conjunto de partículas.

Tercero, la nueva posición para cada partícula es calculada considerando la nueva velocidad asociada y se obtiene empleando la ecuación (2.6)

$$x^i_{k+1} = x^i_k + v^i_{k+1} \tag{2.6}$$

En la figura (2.2) se muestra un gráfico vectorial que representa el cambio de posición de una partícula. Los vectores representados en línea continua son las magnitudes de cada vector inicial. Los vectores representados en segmentos de línea son vectores que contribuyen al cálculo de la nueva posición ponderados por los coeficientes w_p, $c1$ y $c2$, que representan el coeficiente de inercia para la velocidad y los coeficientes cognitivo y social de cada partícula de manera respectiva.

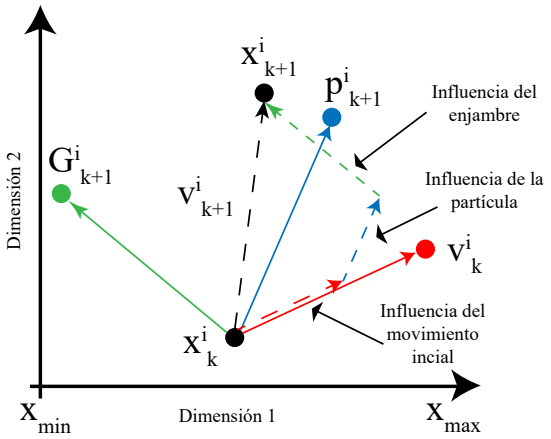

Figura 2.2: Cálculo de la nueva posición de una partícula

El algoritmo (2), corresponde a la implementación del optimizador por enjambre de partículas. Las líneas de código generadas son presentadas en el apéndice A.

2.4. Colonia Artificial de Abejas (ABC)

2.4.1. Introducción de ABC

En la naturaleza, los insectos se comportan bajo estructuras sociales, formando comúnmente colonias; como ejemplo, tenemos a las abejas, avispas, termitas, etc. Dentro de las colonias las estructuras sociales o castas se encuentran bien establecidas, por lo tanto, cada individuo tiene una tarea específica para cada actividad que es necesaria desarrollar en el enjambre [36]. Como ya hemos planteado, el comportamiento de enjambre se caracteriza por la autonomía, el funcionamiento distribuido y la autoorganización. Los sistemas de comunicación entre los insectos de manera individual, contribuyen al patrón de inteligencia colectiva, que se ha determinado nombrar como *"Inteligencia de Enjambre"*.

Algorithm 2 PSO

PSO: $(N_p, X_i, x_{min}, x_{max}, Iter, V_i, v_{min}, v_{max}, c_1, c_2, w_p,)$ parámetros iniciales;
N_p **:** Número de partículas,
$X_i(i = 1, 2, .., N_p)$**,** **:** Posición de las partículas distribuidas de manera uniforme en el espacio de búsqueda,
V_i **:** Velocidad inicial de las partículas,
c_1, c_2, w_p **:** Coeficientes personal, social e inercial, respectivamente,
$Iter$ **:** Número máximo de iteraciones;
Begin
    Evaluar cada partícula en la función objetivo;
    Determinar el mejor valor de posición y velocidad global p_{best}
    Determinar el mejor valor de la función objetivo global G_{best};
    while $k < Iter$ **do**
        for $i = 1 : N_p$ **do**
            Actualizar la velocidad empleando la ecuación (2.5)
            Actualizar la posición empleando la ecuación (2.6)
            Evaluar la partícula en la función objetivo
            if $X_i > p_{best}$ **then**
                $p_{best} = Xi$
            end if
            if $X_i > G_{best}$ **then**
                $G_{best} = Xi$
            end if
        end for
        $k = k + 1$
        w_p= $0.9 * w_p$
    end while
End

En la naturaleza, las abejas recolectoras de miel crean nidos que son llamados "colmenas" (se ha determinado un valor promedio de 20,000 individuos dentro de una colmena) [37] y trabajan en un orden social establecido, teniendo diferentes tareas asignadas. Dentro de la estructura de las castas tenemos la clase "Abeja Reina", la clase "Abeja Zángano" y la clase "Abeja Obrera". Para la actividad de recolección de alimento, la clase "Abeja Obrera" tiene una subdivisión que corresponde a "Abeja Recolectora", "Abeja Observadora" y "Abeja Exploradora". El algoritmo nombrado colonia artificial de abejas (del término inglés : "Artificial Bee Colony", ABC) contempla reproducir el comportamiento de las abejas al desarrollar la tarea de recolección de alimento.

2.4.2. Desarrollo de ABC

El algoritmo de abejas, ABC, fue propuesto por Dervis Karaboga en el año 2005 [38,39]. Este algoritmo se basa en el comportamiento organizacional de las abejas en la tarea de la búsqueda

de alimento, para esta actividad se tienen abejas recolectoras, observadoras y exploradoras. En el proceso de optimización, se considera que el número de abejas obreras es igual al número de fuentes de alimentos, de donde, una fuente de alimentos se considera como una propuesta de solución y su pertinencia es obtenida al ser evaluada en la función objetivo. Dada una fuente de comida o propuesta de solución, las abejas obreras buscan alrededor una mejor calidad del alimento; esto equivale a buscar una nueva solución alrededor de la propuesta inicial. Cada nueva propuesta es evaluada y comparada con la solución anterior; esta información recabada por las abejas obreras es transmitida a las abejas observadoras, las cuales, se encargan de seleccionar las fuentes de alimentos de mejor calidad.

Después de algún número de intentos establecidos, si no se logra mejorar la calidad de la comida o de la solución numérica, la fuente de alimentos es desechada y la abeja obrera asociada a esa fuente de alimento, se convierte en una abeja exploradora. En la fase de exploración las abejas buscan de forma aleatoria una nueva fuente de comida o solución.

Este proceso se repite de forma iterativa y el pseudocódigo que la representa de manera general es presentado en el Algoritmo (3) [40].

Algorithm 3 Código Principal ABC

```
1:  Data: Valores iniciales del conjunto de parámetros;
2:  SN: Número de fuentes de alimento,
3:  MCN: Contador de Iteraciones Máximas;
4:  Begin
5:      Inicializa evaluando las fuentes de alimentos
6:      contador = 1
7:      while contador < MCN do
8:          Fase de Abejas Recolectoras (ver Algoritmo 4)
9:          Fase de Abejas Observadoras (ver Algoritmo 5)
10:         Registro en Memoria de la Mejor Solución
11:         Fase de Abejas Exploradoras (ver Algoritmo 6)
12:         contador = contador + 1
13:     end while
14: End
```

Como ya se ha mencionado, la casta de abejas que desarrolla la tarea de la recolección de alimentos, es nombrada como abejas recolectoras. El código del Algoritmo 4, simula de forma iterativa el comportamiento de ellas.

Para el Algoritmo 4, en la línea número cuatro, se hace referencia a la Ecuación (2.7). El término x_{ij}, hace referencia al valor numérico actual de la abeja en la fuente de alimentos i; mientras que el parámetro x_k, nos indica la fuente de alimentos a ser modificada y su elección

es de forma aleatoria, dentro de todas las fuentes posibles. Además, se debe tomar en cuenta el subíndice j que indica dentro de una muestra el rasgo o característica cuyo valor numérico será modificado. Es decir, de la Ecuación (2.7) se modificará la fuente de alimentos k en su característica j, esta característica se determina de manera aleatoria considerando el termino ϕ_{ij}, el cual esta determinado por una distribución uniforme en el rango de $[-1, 1]$.

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{2.7}$$

Algorithm 4 Fase de Abeja Recolectora

```
1:  Data: Número de fuente de alimento;
2:  Begin
3:      for SN < x_i do
4:          nueva solución x' ← aplicar Ecuación (2.7)
5:          f(x') ← evaluar la nueva solución
6:          if f(x') < f(x_i) then
7:              x_i = x'
8:              explorar(x_i) = 0
9:          else
10:             explorar(x_i) = explorar(x_i) + 1;
11:         end if
12:     end for
13: End
```

Para la fase cíclica de la casta de abejas observadoras, se tiene el Algoritmo 5. Para esta fase, las abejas observadoras se encargan de recopilar la información de cada una de las abejas obreras y seleccionar las fuentes de alimento de mayor calidad. Cada fuente de alimento es evaluada con la función de optimización para poder determinar el grado de calidad del alimento proveniente de una fuente. La Ecuación (2.8), es aplicada para determinar la calidad de cada una de las fuentes de alimento de alimento y para ello se considera el cálculo de la probabilidad. Las fuentes de alimento con mayor valor de probabilidad son elegidas para ser nuevamente exploradas.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \tag{2.8}$$

Finalmente, la fase de exploración es analizada en el Algoritmo 6. En esta fase, las abejas recolectoras han abandonado una fuente de alimento debido a que, después de alguna cantidad de ciclos y de explorar alrededor de una fuente de alimentos, no se ha encontrado una mejor calidad. Es decir, la función de optimización, al ser evaluada no ha presentado una mejora en sus valores numéricos. Para la búsqueda de una nueva fuente de alimentos, en una nueva localidad,

Algorithm 5 Fase de Abeja Observadora

Data: Número de fuente de alimento; Probabilidad de cada solución
Begin
  for $SN < x_i$ **do**
    $p_i \leftarrow$ *asignar probabilidad según Ecuación*(2.8)
  end for
  $i = 0$
  $t = 0$
  while $t < SN$ **do**
    $r = rand(0, 1)$
    if $r < p(i)$ **then**
      $t = t + 1$
      $x' \leftarrow$ *una nueva solución, aplicar Ecuación* (2.7)
      $f(x') \leftarrow$ *evaluar la nueva solución*
      if $f(x') < f(x_i)$ **then**
        $x_i = x'$
        exploración $(x_i) = 0$
      else
        exploración (x_i) = *exploración* $(x_i) + 1$
      end if
    end if
    $i = (i + 1)\ mod(SN - 1)$
  end while
End

se emplea la Ecuación (2.7). Este algoritmo, basado en el comportamiento de las abejas, se ha aplicado en este trabajo debido a su baja complejidad en su implementación computacional y a su alto desempeño en la búsqueda de soluciones sucesivas en problemas de optimización [39].

Algorithm 6 Fase de Abeja Exploradora

Data: Número de fuente de alimento; Contador de exploración
Begin
  $si = i : exploración(i) = max(exploración)$
  if *exploración*$(si) >$ *limite de exploración* **then**
    $x_{si} =$ *random nueva solución, aplicar Ecuación* (2.7)
    exploración$(si) = 0$
  end if
End

Los algoritmos (3, 4, 5, 6), corresponden a la implementación completa del algoritmo de colonia artificial de abejas y las líneas de código generadas son presentadas en el apéndice C.

2.5. Ejemplos de Aplicación

La evaluación del desempeño de métodos metaheurísticos ha tomado gran interés recientemente [41–43] y, para esta tarea, se han propuesto diferentes ecuaciones o sistemas de ingeniería en diferentes ámbitos. En este subtema, presentamos tres tareas a resolver con los algoritmos metaheurísticos revisados: la primera tarea, es determinar la velocidad de convergencia para encontrar las coordenadas de un punto óptimo (valor de mínimo global) al evaluar una función de prueba. Como segunda tarea, se aborda el problema de agrupar datos considerando semejanza entre ellos basándose en los valores numéricos de características establecidas. Como tarea final, se aplican los algoritmos propuestos para la búsqueda de los valores óptimos en parámetros que realizan el proceso de mejorar el brillo y contraste en imágenes digitales.

2.5.1. Optimización en Ecuación Matemática

Se han propuesto ecuaciones matemáticas para poder evaluar la velocidad de convergencia de los algoritmos de búsqueda. La tarea es encontrar los valores de los parámetros en la ecuación que la hacen óptima bajo alguna consideración. Estas ecuaciones (también llamadas funciones de prueba), buscan presentar retos de convergencia a los algoritmos a evaluar, por lo que presentan en diferentes posiciones, mínimos locales dentro del espacio de trabajo, sin olvidar que contienen un mínimo global (punto óptimo) (ver Figura 2.3). Además, estas funciones logran que el punto óptimo se encuentre fuera del origen del sistema de coordenadas y que, el valor del mínimo global sea similar a los valores en mínimos locales, es decir, que el punto óptimo no sea un punto profundo dentro del espacio de trabajo. Para evaluar los métodos de búsqueda metaheurísticos revisados, se propone la función de optimización nombrada "EggHolder" [44], debido a que cuenta con múltiples mínimos locales y el punto óptimo se encuentra fuera del origen del sistema coordenado, lo cual, la ha convertido en una función de referencia para la evaluación de algoritmos de búsqueda.

En la ecuación (2.9) *("EggHolder")*, el espacio de solución o búsqueda es de tipo bidimensional (2D) y su representación gráfica se observa en la Figura 2.3. La Tabla 2.2 presenta sus principales características.

$$f(x,y) = -(y+47)\sin\sqrt{\left|\frac{x}{2}+(y+47)\right|} - x\sin\sqrt{|x-(y+47)|} \tag{2.9}$$

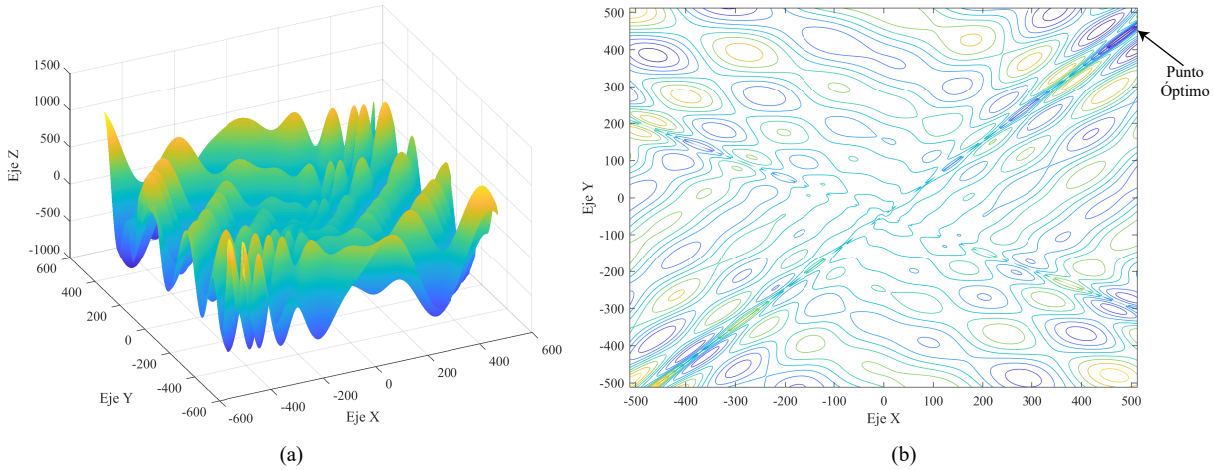

Figura 2.3: Función EggHolder. (a)Vista Isométrica. (b)Vista superior con contornos de nivel.

Tabla 2.2: Parámetros de la Función EggHolder

Nombre	Función Matemática	Mínimo Global	Espacio de Búsqueda
Egg-Holder	Ecuación (2.9)	f (512, 404.2319)=-959.6407	$-512 \leq x, y \leq 512$

2.5.2. Agrupamiento de Datos

El agrupamiento de datos es una tarea que consiste en analizar los valores registrados para un evento y compararlos con otros eventos para determinar si comparten algún rasgo en común; de ser así, estos son agrupados para generar una clase o definir un tipo de evento. Para desarrollar un ejemplo del agrupamiento de datos se han considerado tanto, la Ecuación (2.10) [45] como función objetivo, como la base de datos nombrada "Iris" [46].

La función objetivo (es común en terminología inglesa nombrarla "fitness function") presentada en la Ecuación (2.10), en conjunto con (2.11), de manera iterativa calcula los vectores prototipos de una clase, donde el número de clases es proporcionado como parámetro inicial del proceso de agrupamiento.

$$J_e = \frac{\sum_{j=1}^{N_c} \left[\sum_{\forall Z_p \in C_{i,j}} \frac{d(z_p, m_j)}{|C_{i,j}|} \right]}{N_c} \tag{2.10}$$

$$d\left(z_p, m_j\right) = \sqrt{\sum_{k=1}^{N_d} \left(z_{p,k} - m_{j,k}\right)^2} \tag{2.11}$$

De donde:

- $d(z_p, m_j)$ = representa la distancia de un dato con respecto a los vectores prototipo de cada clase y es definida por la Ecuación (2.11).

- $|C_{i,j}|$ = es la cantidad de vectores de datos que pertenecen a la clase $C_{i,j}$.
- N_c = representa el número de clases.
- N_d = representa el número de dimensión de los vectores de datos.
- z_p = representa el $z - ésimo$ vector de dato.
- m_j = representa el centro o vector prototipo de una clase.

Una base de datos representativa para el agrupamiento de datos es la base de datos Iris [46], la cual es quizás, la base de datos más empleada en el análisis de agrupamiento de datos y reconocimiento de patrones, fue presentada por primera vez por el investigador Ronald Fisher para evaluar un sistema de clasificación lineal [47]. Es una base de datos que contiene 50 muestras de tres tipos de *"Flor Iris"*, que son nombradas como *"Iris Setosa"*, *"Iris Virginica"* e *"Iris Versicolor"*. Los atributos principales de la base de datos son:

- Largo del Sépalo en centímetros.
- Ancho del Sépalo en centímetros.
- Largo del Pétalo en centímetros.
- Ancho del Pétalo en centímetros.
- Clases = 1.-Setosa, 2.-Virginica, 3.-Versicolor.

El gráfico presentado en la Figura 2.4 presenta la distribución de los datos recopilados de la base de datos Iris.

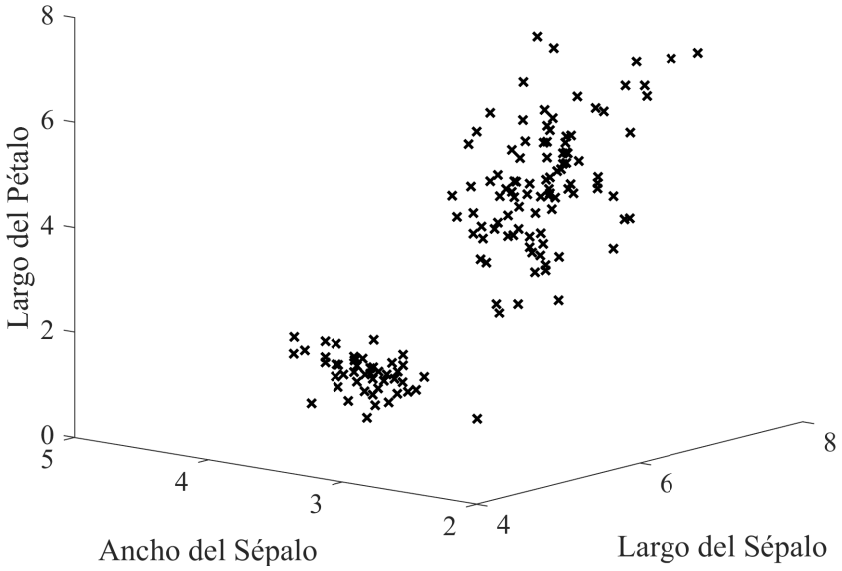

Figura 2.4: Distribución de Datos Iris

2.5.3. Procesamiento de Imagen

En el proceso de captura de imágenes, una problemática común es el cambio de iluminación lo cual provoca imágenes de bajo contraste, por lo tanto, la tarea es mejorar el contraste por procesos matemáticos preparando las imágenes para un segundo proceso. En el proceso de obtención de imágenes médicas de fondo de ojo [48–50], por la técnica empleada en su captura, que constan de dilatar la pupila, es necesario contar con una habitación que presente baja iluminación para protección del paciente; esto provoca que las imágenes tiendan a un bajo contraste y sean oscuras. En este tipo de imágenes es necesario realizar una mejora en el contraste ayudando, con ello, a los médicos a realizar un mejor diagnóstico. Para esta tarea se emplean los algoritmos propuestos en este trabajo, los cuales determinan el juego de parámetros requeridos para realizar el proceso de mejora de imágenes en su característica de contraste. Para cuantificar el contraste se ha contemplado la medición del índice de difusividad, ya que un valor mayor del índice de difusividad indica un mejor contraste.

Como primer paso, se considera la trasformación de la imagen al plano difuso, esta trasformación se realiza empleando la Ecuación (2.12) que desarrolla la función de transformación tipo "S", como función de pertenencia, en la cual el conjunto de parámetros [a, b, y c] determina la forma de "S". La problemática a resolver es encontrar los valores adecuados de estos parámetros [a, b, c] para lograr un mayor índice de difusividad. En la Figura 2.5 se presenta la gráfica de la función "S"para los valores de $a = 50, b = 128, c = 200$.

$$S(a,b,c) = \begin{cases} 0 & x < a \\ 2 \times \left(\frac{x-a}{c-a}\right)^2 & a \leq x \leq b \\ 1 - 2 \times \left(\frac{c-x}{c-a}\right)^2 & b \leq x \leq c \\ 1 & x \geq c \end{cases} \tag{2.12}$$

El cálculo del índice de difusividad se desarrolla empleando la Ecuación (2.13).

$$\gamma(X) = \sum_{k=0}^{L-1} \text{mín}\left[\mu_x(k), 1 - \mu_x(k)\right] h(k) \tag{2.13}$$

De donde:

- L = indica el máximo nivel de gris L, que corresponde al blanco.
- $h(k)$ = indica el número de ocasiones que se presenta el nivel de gris k en el histograma de la imagen.

- $\mu(k)$ = indica el valor de pertenencia del nivel de gris k y es calculado empleando la función "S".

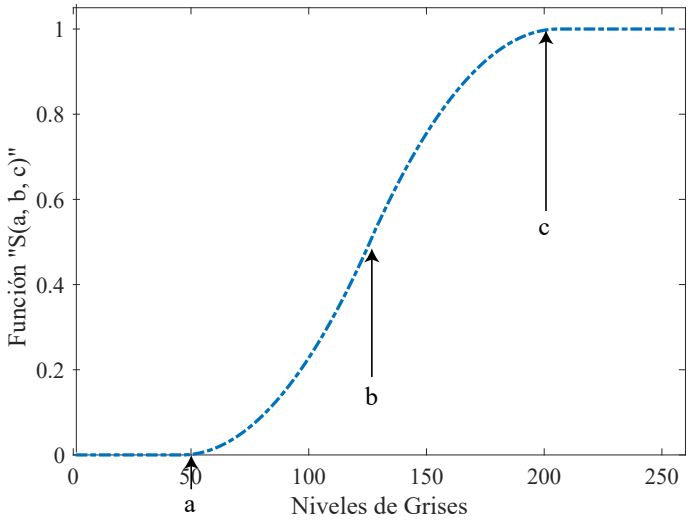

Figura 2.5: Función S, para a=50, b=128 y c=200

2.6. Análisis de Desempeño de las Tres Técnicas Metaheurísticas

Buscando evaluar el desempeño de los algoritmos metaheurísticos presentados, se han aplicado en las tres pruebas descritas: función matemática, agrupamiento de datos y mejora de contraste en imágenes; buscando determinar con ello el algoritmo de mejor desempeño, para ser considerado en procesos posteriores dentro del trabajo presentado.

La primera prueba se realiza con la función Eggholder, ya que presenta el reto de contener múltiples mínimos locales y un mínimo global fuera del origen de las coordenadas, dentro del espacio de solución. Por lo tanto, los algoritmos metaheurísticos son aplicados para encontrar los valores numéricos de las variables de la función representan el valor mínimo global.

Para el algoritmo basado en el comportamiento de las hormigas [31] (ACO-R), se han considerado 100 hormigas o agentes, con un factor de intensificación de feromona q =0.5, un tamaño de nuevas soluciones $NewS = 40$ y una razón de relación desviación-distancia, $z = 1$.

Los parámetros empleados para el algoritmo de optimización por enjambre de partículas [35] (PSO), considerando la Ecuación (2.5), presentan los siguientes valores: coeficiente de inercia $w = 1$, disminuyendo en cada iteración a una razón de $w_i = w_i{*}0.99$, el coeficiente personal, asociado al mejor valor registrado por una partícula, $c1 = 2{,}0$ y el coeficiente asociado al mejor valor de solución global del conjunto de partículas $c2 = 2{,}0$, y una población $Np = 100$ partículas.

El algoritmo de colonia artificial de abejas [39] (ABC), tiene los siguientes valores en sus parámetros: el valor para el límite de pruebas alrededor de una fuente de alimento $Lim = 100$, una población de abejas recolectoras, observadoras y exploradoras $Arec = Abs = Aexp = 100$.

Para los tres algoritmos, se han ejecutado 100 ocasiones y por cada ocasión se realizaron 500 iteraciones; y se registra el menor y mayor número de iteraciones en el que se encuentra la solución; se calcula el valor promedio empleando el valor de las coordenadas para el punto que corresponde al mínimo global y el registro del valor total de la función Eggholder. Los valores obtenidos son presentados en la Tabla 2.3.

En la Figura 2.6 se muestran las gráficas de desempeño de una iteración donde los algoritmos han logrado converger al valor óptimo. En estas gráficas, el eje "x" se representa el número de iteraciones, mientras que el eje "y", no presenta el valor de la función matemática evaluada con el mejor resultado para cada iteración. De las gráficas se puede observar que después de un número determinado de iteraciones, las variaciones en el valor de la función matemática evaluada son mínimas, sin embargo, los algoritmos cumplen 500 iteraciones en todas las pruebas y se registra el tiempo de ejecución total.

Tabla 2.3: Respuesta de Algoritmos Metaheurísticos: Función EggHolder

	Número de Iteración Mínima	Número de Iteración Máxima	Número de Iteración Promedio	Número Ocasiones Valor Óptimo	Número Ocasiones Valor No Óptimo	Desviación Estandar Valores	Tiempo Computacional 500 Iteraciones
ACOR	12	415	114	91 (-959.6407)	9 (-821.1965)	19.3932	264.158 s
PSO	11	314	50	92 (-959.6407)	8 (-718.1675)	21.793	51.378 s
ABC	9	381	162	**95** (-959.6407)	**5** (-958.5077)	**1.1524**	132.267 s

Basándonos en los valores de la desviación estándar, el número de ocasiones de convergencia al valor óptimo y el número de ocasiones que el algoritmo falla en localizarlo, presentados en la Tabla 2.3, se concluye para este problema que, el algoritmo de colonia artificial de abejas tiene el mejor desempeño.

Para la segunda prueba, se realizó el proceso del agrupamiento de datos sobre una base de datos típica nombrada *"Base Datos de Flores Iris"*, [46]. Para esta base de datos se propuso como valor inicial una partición en dos clases ($Clases = 2$) y el uso de la Ecuación (2.10), como función objetivo. Los valores de los parámetros para los tres algoritmos son almacenados y la Tabla 2.4 resume los resultados.

Analizando los datos de las Tabla 2.4, podemos mencionar que los tres algoritmos mantienen

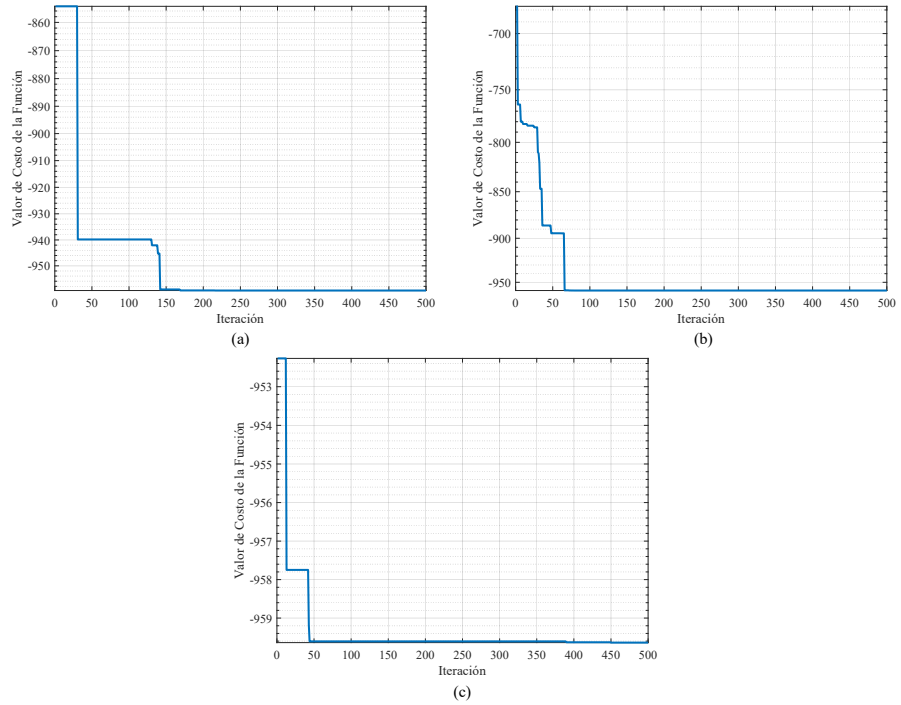

Figura 2.6: Función EggHolder. (a)Respuesta ACOR para caso Óptimo. (b)Respuesta PSO para caso Óptimo. (c)Respuesta ABC para caso Óptimo.

un desempeño sobresaliente y logran separar los datos en dos clases, como se observa en la Figura 2.7. Sin embargo, de acuerdo a los resultados obtenidos, el algoritmo de colonia artificial de abejas es el que presenta nuevamente el mejor desempeño. En la Figura 2.8 se presentan las gráficas de desempeño de los tres algoritmos para una corrida exitosa para la clasificación de los datos.

Para la tercera prueba, se ha considerado la mejora de iluminación en imágenes de fondo de ojo, donde se considera aplicar como función de optimización, el nivel de difusividad en la imagen, ya que nos indica con un bajo valor numérico, que los píxeles tienen definida la pertenencia a una zona determinada (cuando el nivel difuso es cero, no hay confusión en la pertenencia de un píxel a una región), por lo tanto, en la imagen se encuentran zonas de iluminación uniforme, convirtiendo esta técnica en una alternativa para evaluar la calidad de una imagen. Los algorit-

Tabla 2.4: Respuesta de Algoritmos Metaheurísticos: Cluster

	Número de Iteración Mínima	Número de Iteración Máxima	Número de Iteración Promedio	Número Ocasiones Valor Óptimo	Número Ocasiones Valor No Óptimo	Desviación Estandar Valores	Tiempo Computacional 500 Iteraciones
ACOR	88	155	112	95 (0.71401)	5 (0.71633)	0.005115	828.1762 s
PSO	18	382	134	94 (0.71401)	6 (0.71915)	0.008176	751.9733 s
ABC	27	292	105	**97** (0.71401)	**3** (0.71461)	**0.000381**	916.6618 s

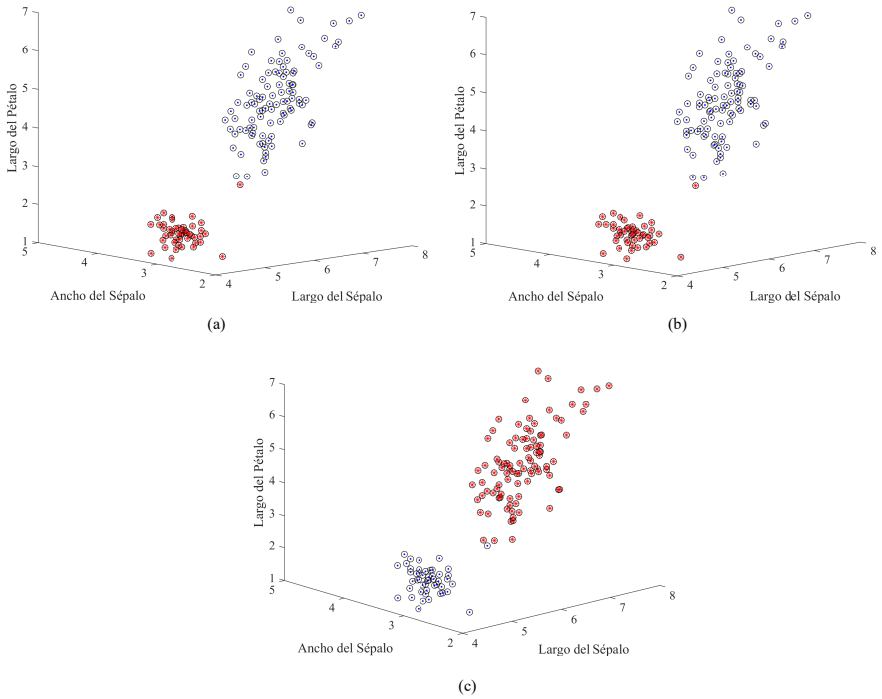

Figura 2.7: Vista Isométrica de la Clasificación de los Datos. (a) Clasificación ACOR para $Clases = 2$. (b) Clasificación PSO para $Clases = 2$. (c) Clasificación ABC para $Clases = 2$.

mos metaheurísticos, nuevamente, tienen que buscar el grupo de valores en los parámetros de la ecuación (2.12) para minimizar el nivel de difusividad.

Los tres algoritmos metaheurísticos fueron evaluados y sus resultados son presentados en la Tabla 2.5; en ella, se presentan los valores de los parámetros a, b, c, que determinan la forma de la función "S". Con la imagen resultante, se calcula el nivel difuso que presentan en su conjunto los valores numéricos de los píxeles. La Figura 2.9 muestra el resultado obtenido al aplicar el proceso a una imagen con baja iluminación de fondo de ojo.

Tabla 2.5: Respuesta de Algoritmos Metaheurísticos: Mejora de Imagen

	Número de Iteración Mínima	Número de Iteración Máxima	Número de Iteración Promedio	Valores Parámetros en S	Nivel Difuso Gamma
ACOR	29	155	102	a = 0.151 b = 0.485 c = 0.892	0.4359
PSO	25	104	87	a = 0.133 b = 0.451 c = 0.909	0.4321
ABC	30	183	119	a = 0.127 b = 0.382 c = 0.817	0.2955

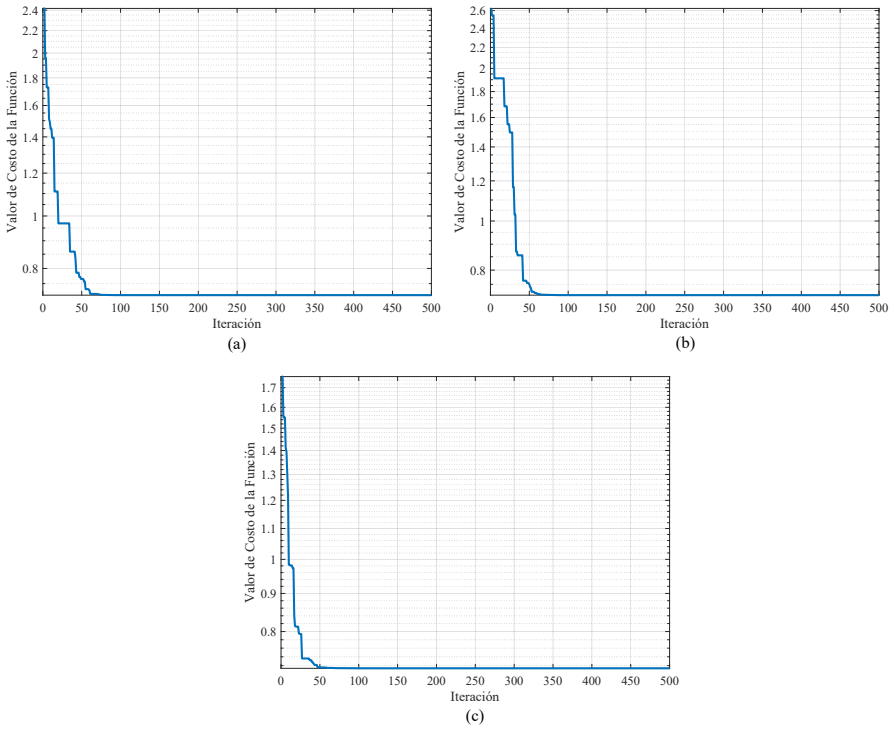

Figura 2.8: Agrupamiento de Datos. (a)Respuesta ACOR para caso Óptimo. (b)Respuesta PSO para caso Óptimo. (c)Respuesta ABC para caso Óptimo.

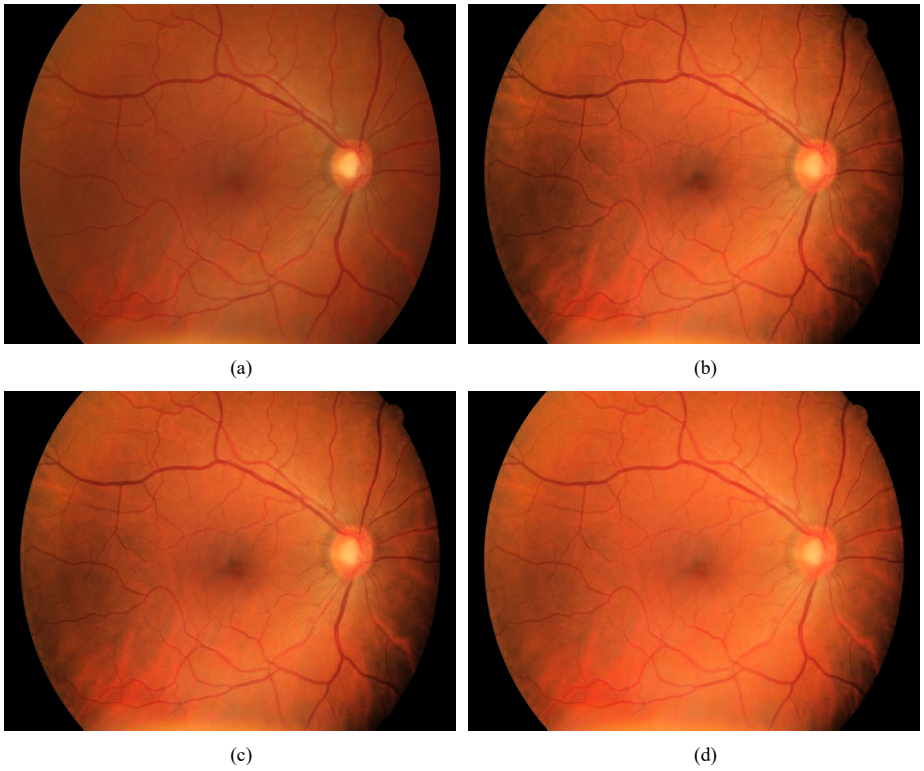

Figura 2.9: Mejora de Imagen. (a) Imagen Original. (b) Imagen con los parámetros obtenidos por ACOR. (c) Imagen con los parámetros obtenidos por PSO. (d) Imagen con los parámetros obtenidos por ABC

2.7. Conclusiones del Capítulo

En el presente capítulo se ha definido el concepto de metaheurística, además, se han tratado los temas referentes a exploración y explotación que desarrollan los entes biológicos en la búsqueda de alimentos y que sirve como base para el desarrollo de los algoritmos de optimización metaheurísticos.

Se han analizados tres algoritmos de optimización inspirados en la naturaleza: Algoritmo de Colonia de Hormigas (ACO), Optimización por enjambre de partículas (PSO) y el algoritmo de colonia artificial de abejas (ABC).

Para determinar el desempeño de los tres algoritmos, se han aplicado a tres tareas: la primera de ellas, es determinar la solución de una función típica en el estudio del análisis de algoritmos de optimización. La segunda tarea, consiste en el agrupamiento de datos considerando la base de datos que contiene muestras de tres características presentes en una familia de flores. La última tarea, es determinar los parámetros de un método de mejora de iluminación en imágenes de fondo de ojo. Con los datos obtenidos de los tres problemas de prueba, se determina que el algoritmo metaheurístico basado en el comportamiento de recolección de alimento de las abejas (ABC), es seleccionado para llevar a cabo los procesos matemáticos de optimización en este trabajo.

3

Redes Neuronales

Las redes neuronales artificiales (RNA), son arreglos de unidades matemáticas que comparten similitudes con las naturales al considerar el mecanismo de excitación, es decir, la neurona recibe las señales que provienen de su entorno, las procesa y genera una nueva señal de salida. Para generar la señal realiza una suma ponderada de las entradas y determina su respuesta basada en un umbral de excitación. De una manera mas generalizada, se menciona que una red neuronal artificial es un sistema de procesamiento de información que tiene ciertas características de desempeño en común con las redes neuronales biológicas (RNB).

Las redes neuronales son sistemas que están diseñados para tratar de modelar la forma en que el cerebro realiza una tarea en particular, sin embargo, aún faltan por determinar de manera concisa los procesos que desarrolla el cerebro para resolver y ejecutar una y múltiples tareas a la vez, de manera tal, que el tema mantiene el interés de los investigadores.

De acuerdo con Davies en [51], las redes neuronales artificiales pueden clasificarse en 3 generaciones. La primera generación, se basa en neuronas propuestas por McCulloch-Pitts basados en unidades de procesamiento simple llamadas *perceptrón* con un umbral de disparo; una característica de estos modelos es que solo pueden responder de manera digital entregando únicamente valores de 0 o 1. Las neuronas de segunda generación están conformadas por estados computacionales, por una suma que representa las entradas sinápticas y una función de activación, también conocida como función de transferencia, que genera la respuesta de la neurona. La función de activación puede ser de tipo lineal, sigmoidal, hiperbólica u otra, para definir un rango de valores como salida. Las redes neuronales de tercera generación son las que mayormente se aproximan al comportamiento de las redes neuronales biológicas. Estas redes neuronales ocupan modelos para reproducir los trenes de pulsos que son empleados comúnmente por las neuronas biológicas para poder comunicarse entre ellas.

3.1. Redes Neuronales de Primera Generación

Martin T. Hagan en su libro *"Neural Network Design"* [52], presenta una recopilación de modelos matemáticos que intentan interpretar los mecanismos de respuestas de las neuronas naturales. Para realizar el proceso de adaptación de las neuronas se realizan cálculos matriciales basados en propuestas de arquitecturas de conexiones de las neuronas.

Los procesos iterativos matemáticos que se realizan para que una arquitectura neuronal pueda desarrollar el aprendizaje o interpretación de una tarea se le conoce como algoritmo de aprendizaje. En los últimos años se han propuesto diferentes técnicas para desarrollar estos algoritmos de aprendizaje. Los algoritmos de aprendizaje optimizan la búsqueda de los pesos sinápticos que determinan la respuesta de la arquitectura neuronal, es decir, se buscan los pesos sinápticos que hacen que la arquitectura neuronal responda de una manera esperada ante una señal de excitación conocida. El algoritmo de aprendizaje sobre redes neuronales artificiales, fue propuesto por los investigadores, McCoulloch-Pitts en [53], donde interpretaban la forma iterativa en que se podían ajustar los parámetros del perceptrón para que éste respondiera de una manera establecida ante una señal de excitación conocida, logrando desarrollar el tipo de respuesta de una función lógica, *"AND"*.

3.1.1. Red Neuronal de Una Sola Capa

En la Figura 3.1, se presenta la arquitectura básica de una RNA de una sola capa (el término de una sola capa hace referencia a tener una sola neurona procesando los datos a su entrada y generado una salida independiente). La neurona artificial, en cada uno de sus nodos de entrada está conectada por medio de elementos de ponderación, conocidos como pesos sinápticos *"w"*. Es interesante notar la presencia de un nodo extra conocido como nodo de polarización, que de manera general tiene el valor de entrada 1 y su ponderación *"b"*; esta entrada extra representa el umbral de disparo de la neurona. Todas estas entradas convergen a un punto suma y el valor obtenido es procesado por una función de activación, generalmente de tipo no lineal, obteniéndose de esta manera la respuesta en relación a una señal de entrada conocida. La función de activación típica para obtener el valor de salida de una neurona *perceptrón* es nombrada como *límite acotado* (en terminología inglesa:*Hard Limit*) y está definida por la ecuación (3.1).

3.1.2. Perceptrón

La neurona *perceptrón* es considerada como la unidad matemática más simple de las redes neuronales, ya que para obtener su salida se consideran unicamente la suma ponderada de dos factores $W.p$ y b. La entrada a la neurona es un vector $\bar{p}$ que contiene las diferentes características o rasgos de un patrón o señal de entrada. Este vector es multiplicado por el vector conformado con los pesos sinápticos $\bar{w}$, (para una sola neurona se debe considerar el vector w' transpuesto para desarrollar la multiplicación entre dos vectores). Si consideramos el caso generalizado para múltiples neuronas se tendrá la matriz de pesos sinápticos $W_{S,R}$ de donde R representa el número de características en la entrada y el valor S la cantidad de neuronas de la red neuronal, por lo tanto, la salida total de la neurona se obtiene aplicando la Ecuación (3.2)

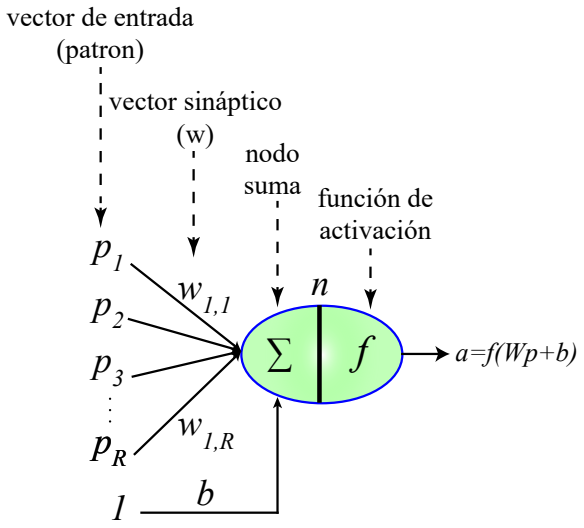

Figura 3.1: Neurona Perceptrón

$$f = \begin{cases} a = 0 & si\ n < 0 \\ a = 1 & si\ n \geq 0 \end{cases} \tag{3.1}$$

$$a = f\left(W \cdot \bar{p} + b\right) \tag{3.2}$$

Para poder desarrollar el proceso de aprendizaje de la neurona perceptrón es necesario proponer valores iniciales para el vector sináptico; lo habitual es generarlo de manera aleatoria en un rango [0,1]. Otro punto a considerar, es que se debe conocer el valor o estado que se pretende alcanzar con la neurona, el cual es nombrado como valor objetivo (de la terminología inglesa: *"target"*). La ecuación de aprendizaje (también nombrada regla de aprendizaje) de la neurona

perceptrón contempla la actualización del vector de pesos sinápticos; para la actualización es necesario calcular el valor del error, éste se obtiene de restar el valor *"target"* y el valor de la salida actual de la neurona, por lo tanto, se presentan dos escenarios: el primero, un error positivo, cuando el valor del target es mayor al valor de la salida actual de la neurona, y el segundo, un error negativo, cuando el valor del target es menor al valor de la salida actual de la neurona. Esto se expresa en la Ecuación (3.3), de donde podemos definir que, los pesos sinápticos nuevos w^{nueva} se obtienen al sumar a los pesos sinápticos anteriores $w^{anterior}$, el valor del error obtenido al considerar el vector de entrada $\bar{p}$.

$$w^{nueva} = w^{anterior} + error \cdot \bar{p} = w^{anterior} + (target - a) \cdot \bar{p} \tag{3.3}$$

El algoritmo (7), presenta el proceso iterativo de la actualización de los pesos sinápticos aplicando la regla de aprendizaje, este proceso comúnmente es nombrado como *"entrenamiento de la red neuronal"*. La red perceptrón ha demostrado ser eficaz en la tarea de clasificación de datos para sistemas lineales.

Algorithm 7 Perceptrón

Perceptron: (T, P, W) parámetros iniciales;
T : valores de referencia o target,
P : matriz de vectores de entrada o patrones,
W : matriz de pesos sinápticos;
Begin
    while $\exists \bar{p} \in P$ and error $\neq 0$ **do**
        for presentar cada vector $\bar{p}$ a la neurona **do**
            if a$(\bar{p}) = target$ **then**
                salida correcta no aplicar Ecuación (3.2)
            else
                calcular $error = target - a(\bar{p})$
                actualizar peso sináptico con Ecuación (3.2)
            end if
        end for
    end while
End

3.2. Redes Neuronales de Segunda Generación

Las redes consideradas de segunda generación son las redes multicapas; surgieron debido a la limitante que se tiene en redes de una sola capa, de tan solo poder resolver problemas de tipo lineal. Una red multicapa puede ser considerada como un aproximador de funciones universal, ya

que para seguir o imitar una función con curvaturas, uniendo segmentos de rectas, considerando que en la primera capa un conjunto de neuronas puede generar múltiples líneas y éstas son procesadas y conjuntadas por una segunda capa de neuronas, logrando de esta manera tener un comportamiento que corresponden a sistemas de tipo no-lineal.

3.2.1. Red Neuronal Multicapa

Una red neuronal se considera multicapa si tiene al menos dos neuronas conectadas en cascada; la salida o respuesta de una neurona se convierte en la entrada de una segunda [52]. Esto se observa en la Figura 3.2 y en ella, el superíndice hace referencia a la capa neuronal, por lo general se emplea la misma función de activación en las neuronas de la misma capa. La capa final de la red es nombrada como *"capa de salida"* y todas las posibles capas anteriores como *"capas ocultas"*. Con una arquitectura neuronal multicapa se extiende el campo de aplicaciones de las redes neuronales artificiales, ya que se pueden resolver problemas de tipo lineal y no-lineal. Recordemos que con una red multicapa, puede desarrollarse un seguidor de funciones universal, las primeras capas pueden generar líneas y las segundas pueden procesarlas y unirlas a manera de poder imitar la forma de una curvatura dada. Con la nueva arquitectura multicapa se generó una nueva regla de aprendizaje para la actualización de la matriz de pesos sinápticos, la cual se nombró propagación del error en retrocesos, pero es mayormente citada como *"BackPropagation"*, por su terminología inglesa.

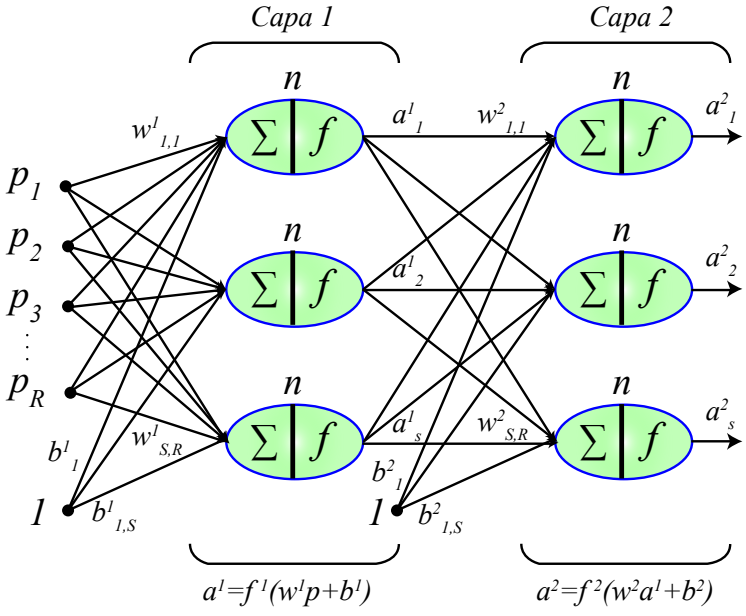

Figura 3.2: Arquitectura Neuronal Multicapa

3.2.2. Técnica de Aprendizaje Backpropagation

En 1986 Rumelhart, presenta el algoritmo de aprendizaje *"BackPropagation"* en [54]; con este algoritmo se puede resolver el problema del entrenamiento en redes neuronales multicapa. El aporte de este algoritmo se basa principalmente en que provee de un método sistematizado para determinar el error en los nodos de las capas ocultas. Cuando el error es calculado en todos los nodos de la red, se vuelve posible actualizar todos los pesos sinápticos logrando de esta manera la convergencia del arreglo de neuronas artificiales.

El algoritmo inicia presentando un vector patrón en la entrada de la red, y la información que genera viaja a través de la capa de entrada, las capas intermedias (capas ocultas) y finalmente a la capa de salida provocando un resultado; este resultado se compara con el valor esperado (target) y se obtiene el valor del error. Este proceso de transmisión de los datos de la entrada hacia la salida es conocido como *"FeedForward"*. En contraste, el algoritmo de "BackPropagation" inicia con el valor del error desde la capa de salida y es propagado hacia atrás por las capas intermedias hasta llegar a la capa de entrada, de allí su nombre *"BackPropagation"*. En comparación, los dos procesos son similares, incluso en la forma en que se propagan las señales; para el primer caso "Feed Forward", el vector de entrada es multiplicado por los pesos sinápticos y para el segundo, es el error quien multiplica a los pesos sinápticos, la única diferencia radica en que las señales de entrada y salida transitan en direcciones opuestas [55].

Matemáticamente, el algoritmo *"BackPropagation"* puede ser sintetizado de la siguiente manera (tomado de [52]) y el primer paso es presentar a la arquitectura neuronal el vector patrón y desarrollar el proceso de *"FeedForward"*:

$$a^0 = \bar{p}, \tag{3.4}$$

$$a^{m+1} = f^{m+1}\left(W^{m+1}a^m + b^{m+1}\right), \ for\ m = 0, 1, ..., M-1\ , \tag{3.5}$$

$$a = a^M. \tag{3.6}$$

El siguiente paso es calcular el error y propagarlo hacia las capas anteriores, con esta propagación se actualizan los pesos sinápticos.

La sensibilidad al error en cada capa, hace referencia al grado en que el error varía en cada capa y esta variación se debe a la presencia de diferentes vectores patrones que definen a diferentes grupos de datos. El valor de sensibilidad al error se obtiene aplicando las Ecuaciones

(3.7) y (3.8). El símbolo *"M"* hace referencia al número total de capas dentro de la arquitectura neuronal y el símbolo *"m"* indica el número de capa especifica sobre la cual se encuentra actuando la ecuación.

$$s^M = -2\dot{F}^M\left(n^M\right)(target - a), \tag{3.7}$$

$$s^m = \dot{F}^M\left(n^m\right)\left(W^{m+1}\right)^T s^{m+1},\ for\ m = M-1, ..., 2, 1. \tag{3.8}$$

Como último paso se presentan las Ecuaciones (3.9) y (3.10), que son empleadas para actualizar los pesos sinápticos y las polarizaciones para cada neurona en cada capa.

$$W^m(k+1) = W^m(k) - \alpha s^m\left(a^{m-1}\right)^T, \tag{3.9}$$

$$b^m(k+1) = b^m(k) - \alpha s^m. \tag{3.10}$$

Algorithm 8 BackPropagation

BackPropagation: (T, P, W, ε) parámetros iniciales;
T : valores de referencia o target,
P : matriz de vectores de entrada o patrones,
W : matriz de pesos sinápticos,
ε **:** valor mínimo de error que se pretende alcanzar;
Begin
    while $\exists \bar{p} \in P$, error $<= \varepsilon$ **do**
        for $\exists \bar{p}$ presentar a la neurona **do**
            Proceso *"FeedForward"*
                aplicar Ecuación (3.4)
                aplicar Ecuación (3.5)
                aplicar Ecuación (3.6)
            calcular $error = target - a(\bar{p})$
            Proceso *"BackPropagation"*
                aplicar Ecuación (3.7)
                aplicar Ecuación (3.8)
                aplicar Ecuación (3.9)
                aplicar Ecuación (3.10)
        end for
    end while
End

3.3. Redes Neuronales de Tercera Generación

Las redes neuronales artificiales desde sus inicios han tenido el objetivo de reproducir el comportamiento biológico, para lo que se han usado modelos muchos más sencillos para describir

ese comportamiento biológico, tratando de simular el proceso de comunicación vía sinapsis y la plasticidad sináptica con la que se modifican las relaciones existentes entre ellas.

Como se ha mencionado, las redes neuronales de tercera generación son las que mayormente se aproximan al comportamiento de las redes neuronales biológicas y los modelos matemáticos propuestos para este tipo de neuronas, intentan reproducir los *"pulsos"* o *"spikes"* que se sabe, generan las neuronas biológicas al ser excitadas [56].

El estudio de estas redes surgió debido a que en los últimos años la evidencia acumulada indicaba que las neuronas biológicas usaban el tiempo de potenciales de membrana o spikes para codificar y procesar la información [57].

3.3.1. Neuronas Pulsantes (Spiking)

El modelo de neurona pulsante *"Spiking"*, tiene sus bases en la interpretación de los mecanismos biofísicos responsables de generar la actividad en las células neuronales. Para proponer un modelo matemático análogo se inicia con una representación biológica y eléctrica de una célula neuronal.

En la Figura 3.3 se muestra un análogo eléctrico de los componentes en una neurona [58]; se tienen signos positivos y negativos alrededor de la membrana celular que representan las cargas dentro y fuera de la membrana celular. Además, se representa la resistencia y la capacitancia asociada a la membrana celular y finalmente, se tiene la representación de la inyección de una corriente externa, por medio de un electrodo representado por I_e.

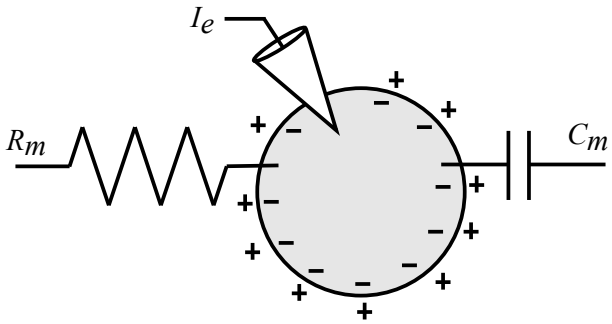

Figura 3.3: Representación eléctrica de una célula neuronal. Adaptada de [58]

En este arreglo la capacitancia y resistencia total de la membrana celular son representadas por C_m y R_m respectivamente. La constante de proporcionalidad por unidad de área, llamada capacitancia de membrana, está indicada por c_m y su valor típico es $c_m \approx 10\ nF/mm^2$. Por lo tanto, para conocer el valor de C_m se aplica la Ecuación (3.11), considerando el área total de la

membrana (A).

$$C_m = c_m A \tag{3.11}$$

De una forma similar se tiene la constante de proporcionalidad $r_m \approx 1\ M\Omega \cdot mm^2$. La resistencia total de la membrana es calculada con la Ecuación (3.12) .

$$R_m = r_m / A \tag{3.12}$$

En la Figura 3.3 se puede apreciar la inyección de una pequeña corriente constante I_e empleando un electrodo; determinado por la ley de Ohm, se tiene $\Delta V = I_e R_m$ y con ella se puede conocer la actividad eléctrica neuronal aplicando en tiempos regulares determinados la corriente inyectada. Si se deja de suministrar la corriente a la célula llegará al restablecimiento de su estado de relajación o voltaje de equilibrio. De esta manera, podemos conocer la corriente de membrana. Esta corriente en su totalidad está determinada por la apertura y cierre de los canales iónicos de sodio y potasio. La corriente total en la membrana es determinada por la suma de las corrientes debido a la diferencia de la apertura y cierre en los canales iónicos presentes en la interacción con un potencial sináptico. La corriente total presente en la membrana es obtenida al multiplicar i_m por la superficie total de la célula, como se expresa en la Ecuación (3.13).

$$I_m = i_m A \tag{3.13}$$

Los investigadores Hodgey y Huxley, presentaron un modelo donde consideran iones de sodio y potasio principalmente, dejando la posibilidad de la existencia de otros tipos de iones [56]. Teniendo en consideración esto, se tiene la Ecuación (3.14), donde el factor g_i es la conductancia por unidad de área o la conductancia específica debido a los canales iónicos.

$$i_m = \sum_i g_i \left(V - E_i\right) \tag{3.14}$$

En Figura 3.4 se presenta el modelo equivalente eléctrico RC de la membrana. En esta figura se tiene la representación de la capacitancia, la resistencia y la corriente externa inyectada. En la Tabla 3.1 se presentan los valores típicos considerados en el modelo.

Para el circuito RC formado, tiene que considerarse su comportamiento en el tiempo, esto es dado en las Ecuaciones (3.15), (3.16) y (3.17). Con estas ecuaciones se puede determinar el

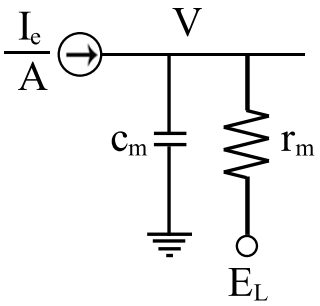

Figura 3.4: Circuito equivalente RC de membrana neuronal. Adaptada de [58]

Tabla 3.1: Parámetros de la Membrana

Parámetro	Valor Típico
c_m	$\approx 10\ nF/mm^2$
r_m	$\approx 1\ M\Omega \cdot mm^2$
C_m	$c_m A$
R_m	r_m/A
E_L	$-70mV$

valor de la corriente dependiente del tiempo. Analizando la Ecuación (3.17), la expresión en el lado derecho se obtiene al considerar las leyes de *Kirchoff* de suma de corrientes. El signo negativo del primer factor indica una corriente inversa con respecto al nodo de análisis V. La parte izquierda representa la corriente derivada de la variación de la capacitancia de membrana en el tiempo.

$$q = C_m V \tag{3.15}$$

$$\frac{dq}{dt} = i = C_m \frac{dV}{dt} \tag{3.16}$$

$$c_m \frac{dV}{dt} = -\frac{(V - E_L)}{r_m} + \frac{I_e}{A} \tag{3.17}$$

$$\tau_m \frac{dV}{dt} = -(V - E_L) + I_e R_m \tag{3.18}$$

Multiplicando por r_m en ambos lados de la Ecuación (3.17) con: $\tau_m = r_m c_m = R_m C_m$ generando la Ecuación (3.18), τ_m es la constante de tiempo de la membrana y juega un rol importante en la determinación de la rapidez con la que reacciona la célula neuronal a los cambios de corriente en la entrada; por lo tanto, τ_m grande, implica una respuesta lenta de la neurona, así mismo con τ_m pequeño, se presenta una respuesta rápida de la neurona. En la Figura 3.5. V_{ss} indica el valor en estado estable que alcanza la neurona ante la presencia

constante de una corriente de entrada. Cuando la corriente externa deja de suministrarse el potencial, desciende a su estado de equilibrio denotado como E_L.

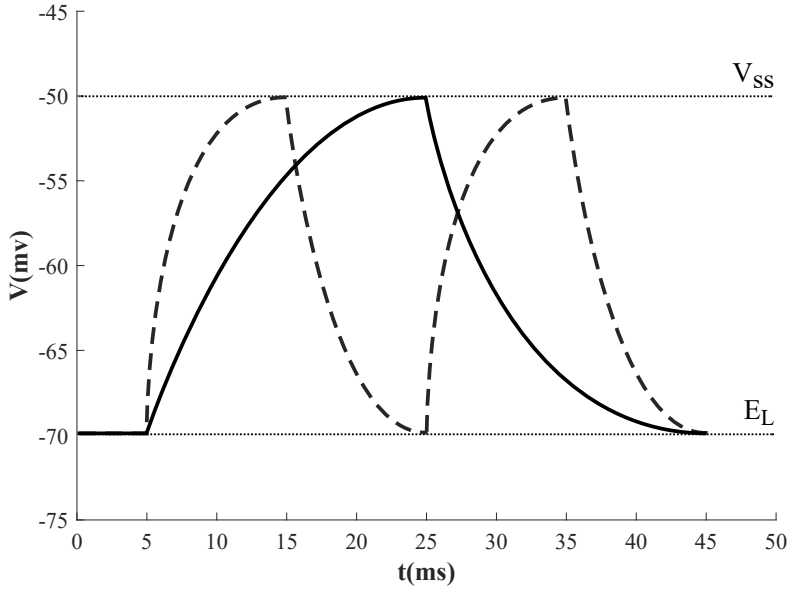

Figura 3.5: Efecto de la constante de tiempo de la membrana τ_m. (a)$\tau_m = 10\,ms$, línea con segmentos. (b) $\tau_m = 20\,ms$, línea continua.

Hasta este momento se ha considerado únicamente el comportamiento de la membrana celular, sin embargo, es importante analizar la presencia de un potencial sináptico y cómo éste modifica el comportamiento celular, por lo tanto, al comportamiento del modelo equivalente RC se le agrega la presencia de la sinapsis. La presencia sináptica es modelada como una conductancia, la cual es expresada en la Ecuación (3.19) y presentada en la Figura 3.6.

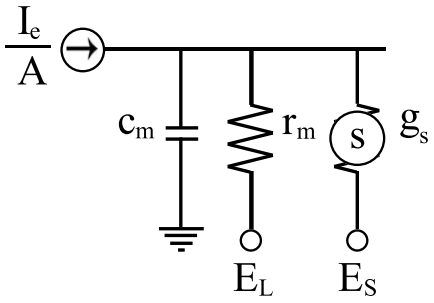

Figura 3.6: Circuito equivalente RC con conductancia sináptica. Adaptada de [58]

$$\tau_m \frac{dV}{dt} = -\left((V - E_L) + g_s\,(V - E_s)\,r_m\right) + I_e R_m \tag{3.19}$$

Es importante determinar en la Ecuación (3.19) el valor de la conductancia sináptica considerando las aperturas de los canales iónicos; para esta condición se considera la probabilidad

de que los canales estén abiertos o cerrados en un tiempo determinado, con lo cual se tiene la expresión en la Ecuación (3.20).

La conductancia asociada a la sinapsis es representada por $g_{s,max}$, P_{rel} es la probabilidad de generar un spike a la salida debido a la presencia de spike en la sinapsis de entrada y P_s es la probabilidad de apertura de los canales ionicos postsinápticos. La relación entre P_s y P_{rel} en función del tiempo se puede ver en la gráfica de la Figura 3.7; en esta gráfica se observa el comportamiento de dos tipos de sinapsis química en el tiempo, *AMPA (a-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor)* y $GABA_a$ *(gamma-aminobutyric acid receptor)*, τ_{peak} es el lapso de tiempo en que se presenta un pulso a la entrada de la neurona y la respuesta máxima de la sinápsis $GABA_a$, en la gráfica se considera $\tau_{peak} = 2{,}7\ ms$. La forma de la sinapsis *AMPA* está expresada por la Ecuación (3.21) y la sinapsis tipo $GABA_a$ por la Ecuación (3.22). La sinapsis tipo *AMPA* se relaciona con disparos rápidos y $GABA_a$ con disparos lentos. Las Ecuaciones (3.21) y (3.22) son modelos que pueden emplearse como generadores de pulsos sinápticos para sistemas computacionales.

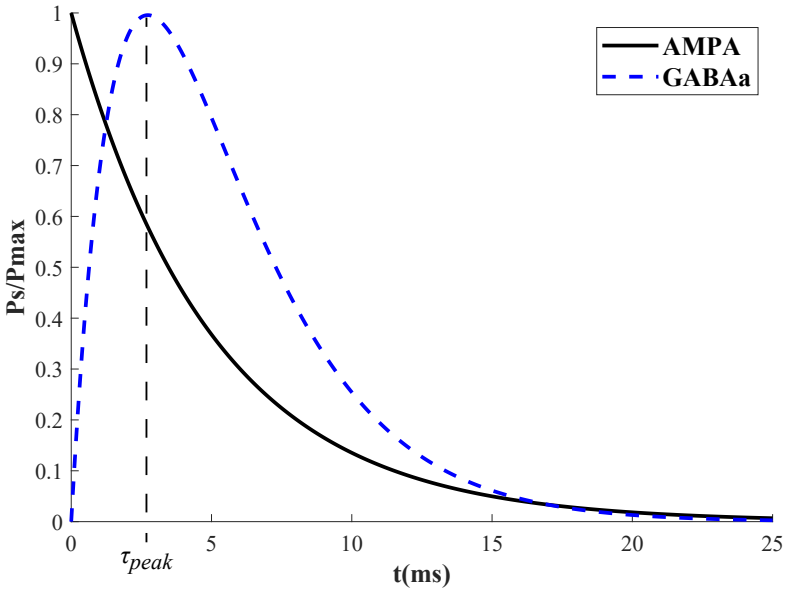

Figura 3.7: Respuesta de la sinápsis química *AMPA* y $GABA_a$.

$$g_s = g_{s,max} P_{rel} P_s \tag{3.20}$$

$$K(t) = \exp^{-\frac{t}{\tau_s}} \tag{3.21}$$

$$\alpha(t) = \frac{t}{\tau_{peak}} \cdot \exp^{\left(1-\frac{t}{\tau_{peak}}\right)} \tag{3.22}$$

3.3.2. Neurona Pulsante de Izhikevich

Izhikevich en [59, 60] propone un modelo neuronal que reproduce los spike de las neuronas corticales; este modelo combina la plausibilidad biológica del modelo Hodgking-Huxley y la eficiencia computacional del modelo *Spike Response Model (RMS)*. Las Ecuaciones (3.23), (3.24) y (3.25), son las que definen el modelo propuesto por Izhikevich.

$$C\frac{\mathrm{d}v}{\mathrm{d}t} = k\,(v - v_r)\,(v - v_t) - u + I_e \tag{3.23}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = a\,(b\,(v - v_r) - u) \tag{3.24}$$

$$Si\ v \geq v_{peak}\ \Rightarrow v = c,\ u = u + d \tag{3.25}$$

De las Ecuaciones anteriores podemos definir los principales parámetros que describen el comportamiento dinámico del modelo neuronal descrito.

- El parámetro a, describe la escala de tiempo de la variable de recuperación u.
- El parámetro b, describe la sensibilidad de la variable de recuperación u, a las fluctuaciones subumbrales del potencial de membrana v.
- El parámetro c, describe el valor de reinicio posterior al pico del potencial de membrana v causado por las conductancias rápidas debido a un umbral alto de iones $K+$ alrededor de la membrana celular.
- El parámetro d, describe el reinicio posterior al pico de la variable de recuperación u, causado por las conductancias lentas debido a un umbral alto de iones $Na+$ y $K+$, que interactúan con la membrana celular.

Además, debido a que reproduce con alta precisión el comportamiento dinámico de una neurona biológica para un amplio conjunto de tejidos cerebrales corticales, el modelo es ampliamente aceptado para diseñar hardware neuromórfico computacional. La Tabla 3.2, presenta los valores de cada parámetro de las configuraciones típicas del modelo neuronal de Izhikevich [59].

La configuración de pulsos regulares *"RS"*, es la que mayormente se presenta en las neuronas del cortex y tiene la peculiaridad de generar pulsos regulares y de amplitudes constantes. Esta configuración es la mas empleada en el diseño de arquitecturas neuronales artificiales pulsantes.

Tabla 3.2: Resumen de los valores de parámetros de las principales configuraciones del modelo neuronal de Izhikevich

Configuraciones Típicas (Acrónimos)	**a**	**b**	**c**	**d**
Regular Spiking (RS)	0,02	0,2	−65	8
Fast Spiking (FS)	0,1	0,2	−65	2
Low-Threshold Spiking (LTS)	0,02	0,25	−65	2
Chattering (CH)	0,02	0,2	−50	2
Intrinsically Bursting (IB)	0,02	0,2	−55	4

En la Figura 3.8, se muestra el gráfico que corresponde a la implementación del modelo matemático de Izhikevich en la configuración de pulsos regulares (Regulas Spiking), tomando los parámetros de la Tabla 3.2. Se puede notar que los pulsos son generados de forma igualmente espaciados y se logran con una corriente constante de $Ie = 100\ picoAmperios$ aplicada durante $t = 1000\ miliSegundos$.

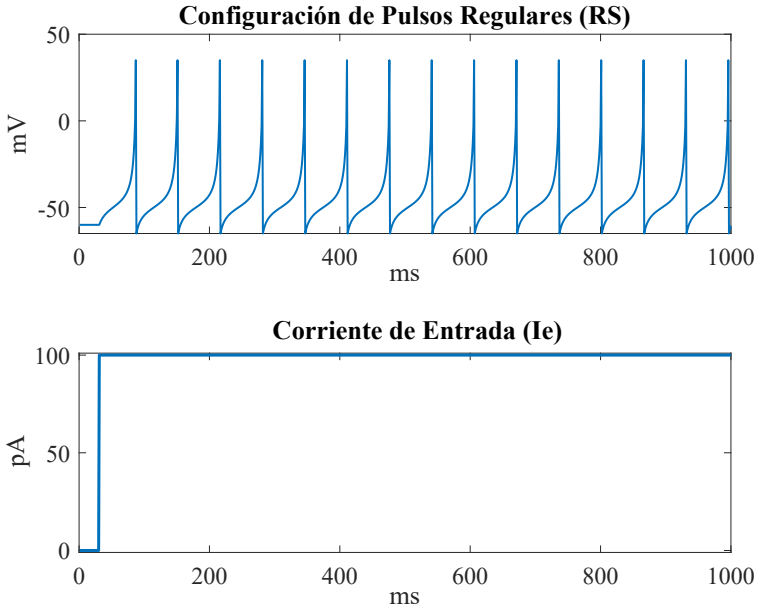

Figura 3.8: Configuración de Pulsos Regulares (RS) del modelo neuronal de Izhikevich

Una característica a resaltar en la configuración *"RS"* es que, a medida que se aumenta la corriente de entrada, de manera lineal aumenta la cantidad de pulsos generados (*frecuencia de los pulsos*). La Figura 3.9, contiene la gráfica de la corriente contra la cantidad de pulsos y se observa que el incremento es casi lineal. En el eje *"x"* se representa la corriente de entrada en picoAmperios y en el eje *"y"* la relación de pulsos por segundo. Es importante notar que antes de los $53pA$, la neurona no tiene la suficiente corriente para generar un pico. Superada la corriente de $53pA$, la neurona artificial genera pulsos regulares; combinando el valor de corriente de entrada con la frecuencia de los pulsos, es posible desarrollar un sistema neuronal que reconoce

una actividad o patrón específico. Se considera esta actividad, en términos de corriente que se suministra a la neurona spiking, ésta responderá a una frecuencia en específico, logrando de esta manera reconocer la actividad presente; esto se aprovecha para reconocer más de una actividad o patrón por una sola neurona, ya que éstas responden a diferentes frecuencias para diferentes niveles de corriente.

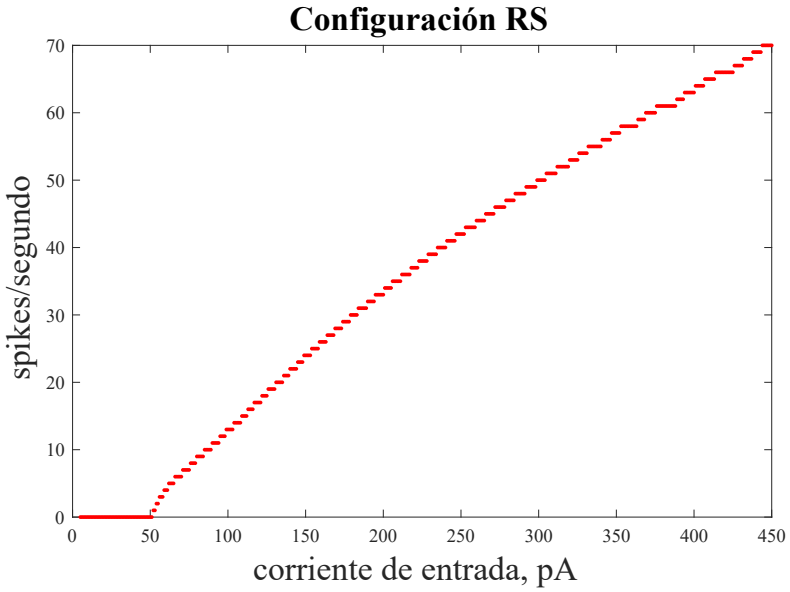

Figura 3.9: Respuesta de una RNS en configuración RS.

3.3.3. Neurona Spiking con Respuesta No Lineal Específica

Si empleamos la configuración *RS* y su variación de corriente contra la frecuencia de pulsos se puede construir un sistema de clasificación lineal con neuronas *Spiking*, pero se presenta nuevamente el problema de trabajar únicamente con problemas en sistemas lineales.

En este trabajo, de una manera original, se ha propuesto generar una salida de tipo no lineal como respuesta de la neurona *Spiking*, considerando que con el juego de valores apropiados de los parámetros $\{a, b, c, d\}$ en el modelo matemático de Izhikevich, se puede lograr una configuración que para niveles bajos de corriente se tenga una salida de frecuencia baja, pero cuando la corriente de entrada se incremente, la frecuencia también se incremente dentro de un rango determinado y, después de un valor establecido de frecuencia, aún cuando la corriente aumente el valor de la frecuencia se mantenga. La configuración propuesta logra presentar un comportamiento de tipo sigmoide y su gráfica es presentada en la Figura 3.10. La señal con segmentos de línea es una función sigmoide de referencia y está descrita por la Ecuación (3.26), mientras que la

señal de puntos y líneas es la respuesta de la *"Neurona Spiking Sigmoidal o NSS"*, propuesta en este trabajo. La finalidad principal de tener un modelo de respuesta no-lineal es dotar a las neuronas *Spiking* con la capacidad de poder procesar problemas de tipo no lineal mejorando de esta manera la versatilidad de ellas.

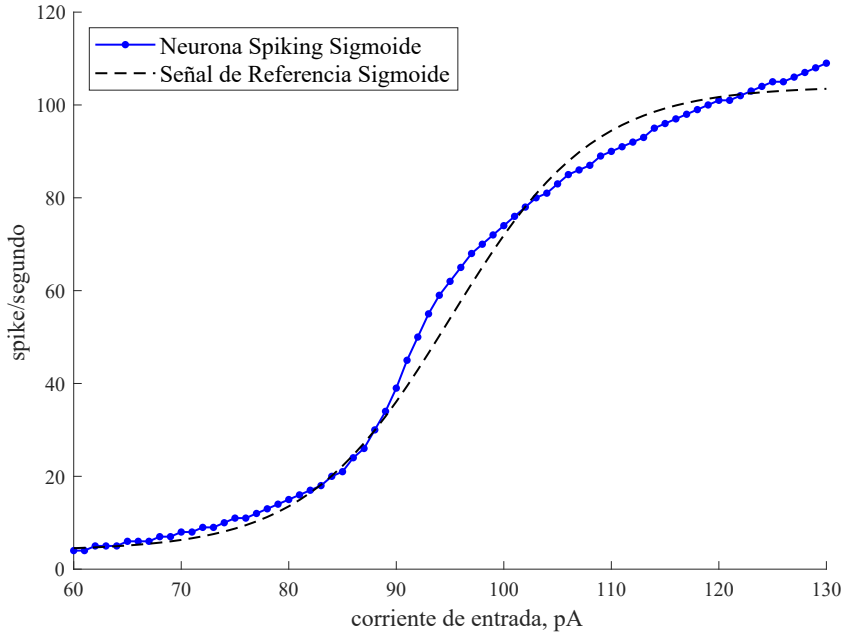

Figura 3.10: Respuesta de una Neurona Spiking Sigmoide.

$$sig = \frac{1}{1 + \exp^{a_{min} \cdot (x - b_{min})}} * 100 \tag{3.26}$$

Para determinar los valores de los parámetros $\{a, b, c, d\}$ del modelo neuronal de Izhikevich, se propone emplear el método metaheurístico de abejas "ABC" y convertir el problema de búsqueda de los valores de los parámetros en un problema de optimización. Para desarrollar esta tarea se considera la Ecuación (3.26) como la ecuación de referencia, de donde los valores *amin* y *bmin*, definen los valores principales de la función sigmoide, es decir, para valores menores a "amin" la respuesta es constante, entre los valores de *amin* y *bmin* se tiene el crecimiento de la función y para valores superiores a *bmin* la salida de la función satura en un valor constante. Posteriormente se toma en cuenta la respuesta del modelo neuronal y se compara con la respuesta de la función sigmoidal, obteniendo de esta comparación el valor del error cuadrático medio, empleando la Ecuación (3.27), donde N_r representa el número de muestras a comparar, y_i la respuesta actual de la NSS, y finalmente y_{obj} son los valores de la ecuación de referencia.

Para aplicar el método de optimización metaheurístico se considera a cada abeja como el

vector de 4 parámetros del modelo neuronal que representa una posible solución; de esta manera se buscará el conjunto de valores que logra minimizar el error cuadrático medio.

$$ECM = \frac{1}{Nr}\sum_{i=1}^{N_r}(y_i - y_{obj})^2 \tag{3.27}$$

Los valores empleados para desarrollar el método de las abejas "ABC" es presentado en la Tabla 3.3 y los valores obtenidos para el modelo de neurona spiking sigmoidal son presentados en la Tabla 3.4, logrando un valor de error cuadrático medio $ECM = 0{,}017$, lo cual indica un alto grado de semejanza entre la señal de referencia y la respuesta obtenida por la *neurona spiking sigmoidal*.

Tabla 3.3: Parámetros del Algoritmo ABC

Parámetro	Algoritmo Metaheurístico ABC
Población	100
Límite de abandono	50
Iteraciones	500
Dimensiones de búsqueda	4

Tabla 3.4: Valores de los parámetros para una respuesta no lineal de una NSS

Parámetro	**Valor**
a	$-0{,}004811\ ms^{-1}$
b	0,196783
c	$-85{,}592772\ mV$
d	$-5{,}343734\ mV$
k	0,9
C	$100\ pF$
v_r	$-60\ mV$
v_t	$-40\ mV$
v_{peak}	$35\ mV$

3.3.4. Clasificación de Sistemas No Lineales con Una Neurona Pulsante

La función lógica XOR, es uno de los problemas clásicos a resolver, muy extendido, ya que ejemplifica la solución de un problema no linealmente separable y aquí es considerado para demostrar lo robusta de una sola neurona spiking al poder resolver este problema. La Tabla 3.5 contiene la respuesta de la función XOR y es considerada como el objetivo a alcanzar con la respuesta de la neurona Spiking.

Los valores de entrada a la neurona Spiking son considerados como niveles de voltaje, los

Tabla 3.5: Tabla de valores para la función lógica XOR

Entradas		Salida
Sp1	Sp2	Spiking
0	0	2 spikes
1	0	3 spikes
0	1	3 spikes
1	1	2 spikes

cuales son multiplicados por las conductancias sinápticas (w); de este proceso se obtienen las corrientes de suministro a la neurona, la cual responde con picos a diferentes frecuencias. La Figura 3.11, muestra la arquitectura formada con una sola neurona. En concreto, para nuestro ejemplo, la neurona considera para su actividad una ventana de tiempo de 1 segundo y genera 2 picos para reconocer las entradas similares ("0, 0 " y "1, 1") y genera 3 picos para entradas combinadas ("0, 1" y "1, 0"); esto se puede observar en la Figura 3.12.

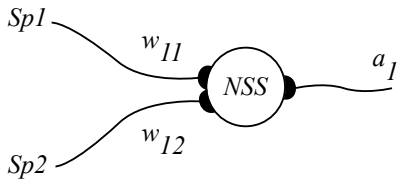

Figura 3.11: Arquitectura con una sola neuronal spiking.

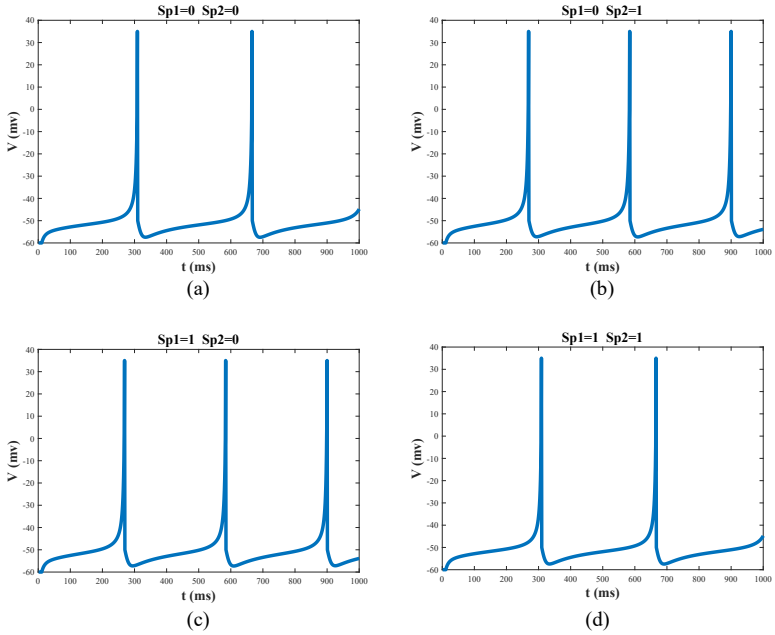

Figura 3.12: Respuesta de la neurona spiking para la función lógica XOR. (a) Para la entrada $S_{p1} = 0$ y $S_{p2} = 0$. (b)Para la entrada $S_{p1} = 0$ y $S_{p2} = 1$. (c)Para la entrada $S_{p1} = 1$ y $S_{p2} = 0$. (d)Para la entrada $S_{p1} = 1$ y $S_{p2} = 1$

Para determinar la respuesta de la neurona Spiking en todo el espacio de trabajo, se realiza un barrido sobre los niveles de voltaje para cada una de las entradas en la neurona, obteniéndose los gráficos en la Figura 3.13, donde se muestran las vistas isométrica a 35° e isométrica superior del espacio de trabajo.

Con lo anterior, se demuestra lo robusto del desempeño de las redes neuronales de tercera generación y la versatilidad que puede tener aún con una sola neurona, convirtiéndolas en una opción viable para el procesamiento de información.

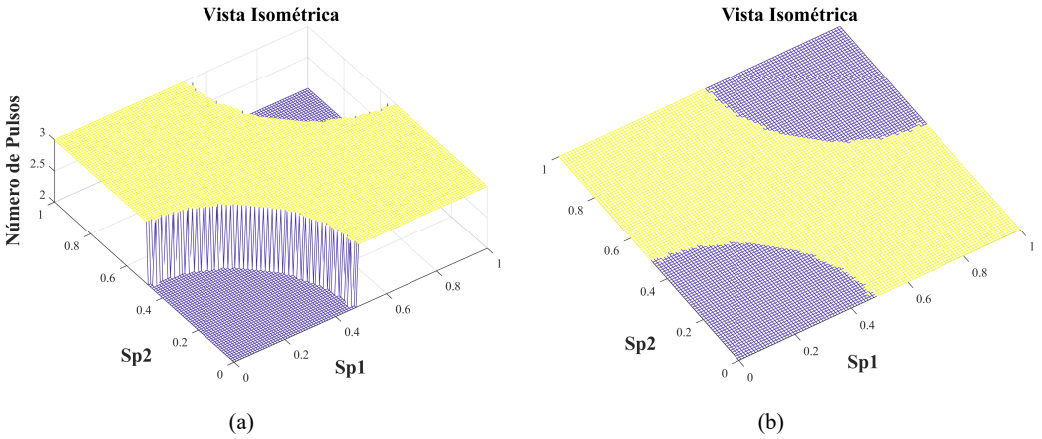

Figura 3.13: Vistas isométricas del espacio de trabajo. (a)Vista a 35°. (b)Vista a 77°.

3.4. Conclusiones del Capítulo

Bajo el concepto histórico del desarrollo de las redes neuronales artificiales, se han presentado las neuronas de primera, segunda y tercera generación. Para cada tipo de neurona se ha presentado su diseño gráfico y el marco matemático del funcionamiento de las mismas. Se ha hecho énfasis en las neuronas de tercera generación, ya que estas desarrollan una mayor semejanza a las neuronas biológicas.

Se ha presentado la sección para el análisis matemático del comportamiento de las neuronas biológicas; con este análisis se ha podido determinar los valores de conductancia, los niveles de corriente y los diferentes tipos de pulsos que se generan en las neuronas biológicas, esto con la finalidad de generar un modelo artificial.

Se ha analizado el modelo matemático presentado por el investigador Izhikevich, del cual se ha propuesto y generado una nueva configuración de respuesta de tipo sigmoide en las neuronas de tipo spiking. Con la respuesta de tipo no lineal de las neuronas spiking, se logra extender su campo de aplicación, ya que es posible abordar problemas de clasificación del tipo mencionado.

4 Internet de las Cosas

El *Internet de las Cosas (IDC)* o también nombrado IoT (de la terminología inglesa *Intenet of Things*), es un concepto que relaciona, el crecimiento en los últimos años, del uso del internet y el registro de datos para monitoreo de actividades en diferentes ámbitos comerciales, industriales e incluso la vida diaria de las personas; de forma breve, se puede decir que es la interconexión de objetos cotidianos a través de internet.

4.1. Introducción al Internet de las Cosas

En el *IDC*, se considera que los objetos tienen integrado un sistema de cómputo numérico dedicado que les ayuda a conectarse al internet. Generalmente un dispositivo IDC, cuenta con sensores que aportan datos del ambiente circundante. Los sensores, actúan para estos objetos, como el equivalente a los sentidos de visión y audición en los humanos que al estar conectados con muchos otros objetos generan redes de información. Como una consecuencia de la concurrencia de esta información; la información recabada puede ser interpretada como patrones de comportamiento, para la toma de decisiones [61, 62]. Un ejemplo de esto en la vida diaria se puede observar en el uso de las pulseras o relojes inteligentes, ya que estos dispositivos cuentan con sensores para determinar la actividad física, las fases de sueño, ritmo cardíaco, entre otras funciones; en ellos se registran las mediciones y generalmente se conectan a dispositivos celulares para presentar un posible diagnóstico del estado de salud del sujeto medido en diferentes lapsos de tiempo.

El concepto de Internet De las Cosas, ha generado conceptos como el de: “Casa Inteligente”, dentro de la cual, los objetos utilitarios son interconectados para conocer los diferentes estados en los que se encuentran diferentes sitios de la Casa, por ejemplo: la TV, el refrigerador y el sistema

de aire acondicionado, y con esto poder evaluar el posible impacto del consumo energético con las actividades de quienes la habitan. Una descripción detallada de la información producida por el conjunto de interacciones de los dispositivos en una Casa Inteligente, puede ser revisada en [63], en donde los protocolos y estándares de comunicación entre ellos son introducidos bajo el concepto de *"Machine to Machine"* o, M2M.

En este sentido, el protocolo *IPv6*, mejora la tasa de envío de datos y tiene un nuevo cifrado en los datos para aumentar la seguridad en la trasmisión. Asimismo, en la generación de nuevos sistemas de almacenamiento de datos fuera de los dispositivos de medición y en muchos casos de los dispositivos de almacenamiento del propio usuario, son nombrados *"Bases de Datos en la Nube"* o simplemente *"Cloud"*.

Como se puede intuir, poco a poco, el Internet de las Cosas gana terreno en nuestras actividades diarias y se proyecta como el sistema que acompañará al género humano en un futuro muy cercano en todas sus actividades.

4.2. Ambientes de Aplicación del Internet de las Cosas

El IDC se considera que aún se encuentra en su etapa inicial, pero con muchas aplicaciones en diferentes campos, desde poder monitorear líneas de producción en una fabrica hasta el monitoreo de jardines. En las ciudades ya se cuenta con el monitoreo de humedad en los jardines públicos, que coordinados con los sistemas meteorológicos determinan el día, la hora y cantidad de agua a suministrar en el riego. Además, en muchas ciudades se tiene el monitoreo de los semáforos, que en conjunto con sensores en el asfalto determinan el flujo en las calles para mejorar la movilidad en ellas. Como es de imaginar, el IDC puede ayudar en la toma de decisiones y a optimizar recursos en las áreas en donde éste se aplique.

4.2.1. Ciudad Inteligente (Smart City)

Las ciudades inteligentes son también el vehículo que transporta el uso del IDC. Los gobiernos en las ciudades han apostado por la recopilación de información y su análisis para la toma de decisiones. Pero si tratamos de definir qué es una ciudad inteligente, se puede mencionar que se trata de una que cuenta con medios electrónicos (redes de sensores, cámaras, estaciones meteorológicas, etc.) cuyos datos son empleados para determinar desarrollos urbanísticos y distribuir de una manera óptima los recursos con los que se cuenta. Un caso específico, en la ciudad

de México, es el monitoreo de la cantidad de personas que hacen uso del sistema de transporte colectivo metro, en sus diferentes modalidades (subterráneo, camiones y teleférico), con lo cual podría ser desarrollado un proyecto de movilidad de personas, determinando las rutas y el tipo de trasporte más eficiente.

4.2.2. Cultivo Inteligente (Smart Farm)

La población en México ha mantenido su crecimiento de forma importante en los últimos años [64] y esto con lleva al aumento de la producción de alimentos para la población; actualmente en México, el paradigma de IDC inicia su aplicación en la agricultura, sin embargo, en otros países ya es una realidad. El IDC se aplica para aumentar la producción en los cultivos y optimizar los insumos empleados en las cosechas. En este ámbito, redes de sensores determinan, humedad, temperatura, luz, salinidad, entre otros parámetros, y esto es complementado en aire, por drones que inspeccionan los cultivos mediante fotografías para determinar la calidad del cultivo en toda una zona. Sobre todo, en cultivos de hoja ancha como lo es el maíz.

4.2.3. Hogar Inteligente (Smart Home)

Uno de los campos donde ha tenido mayor desarrollo el Internet de las Cosas es en el hogar, de hecho, se tiene una definición preliminar nombrada *"Domótica"*, la cual es un sistema con la capacidad de monitorear y automatizar una casa, sin embargo, la domótica al ser conectada a internet se convierte en parte del IDC. En una casa inteligente *(Smart Home)*, se tienen sensores y actuadores para desarrollar tareas dentro del recinto. Se considera que los sensores determinan el estado del ambiente y los actuadores ejecutan las tareas de control, es decir, si los sensores determinan la presencia de lluvia, los actuadores son los encargados de cerrar ventanas y/o puertas.

Las casas inteligentes contemplan principalmente las tareas de:

- Programación de consumo y ahorro energético, al encender o apagar las luces en las habitaciones durante el día o cuando determinan que en una habitación no se encuentra una persona.
- Climatización, al mantener la temperatura constante en una habitación, incluso sin la presencia de una persona, pero con el conocimiento de que está por llegar el residente.
- Como medio de seguridad, se tienen teclados de control de acceso, donde el usuario debe de introducir una combinación numérica y, para aumentar el nivel de seguridad, actualmente

también se registra la forma de escritura de los dígitos para ratificar que es la persona adecuada a quien se la permite el acceso.

- Detectores de humedad, para fugas de agua o presencia de humo y determinar la existencia de un posible incendio.

Con estas medidas y acciones se optimiza el consumo energético y los recursos dentro del hogar, pero principalmente, el Internet de las Cosas, busca el confort de los habitantes.

4.3. Estructura del Internet de las Cosas

El potencial del internet de las cosas va más allá de tan solo conectar dispositivos a internet; para su funcionamiento se tienen que considerar diferentes aspectos como la conectividad y los nuevos paradigmas asociados a ella, como es la implementación y uso de la nueva plataforma de comunicación *5G*, nuevos dispositivos con mayores capacidades de procesamiento y nuevos procesadores en las computadoras. Por lo tanto, podemos definir al internet de las cosas, como un sistema que contiene tres plataformas de desarrollo, interconectadas y trabajando de manera coordinada. El esquema estructural del *IDC* se presenta en la Figura 4.1 y sus componentes son descritos a continuación.

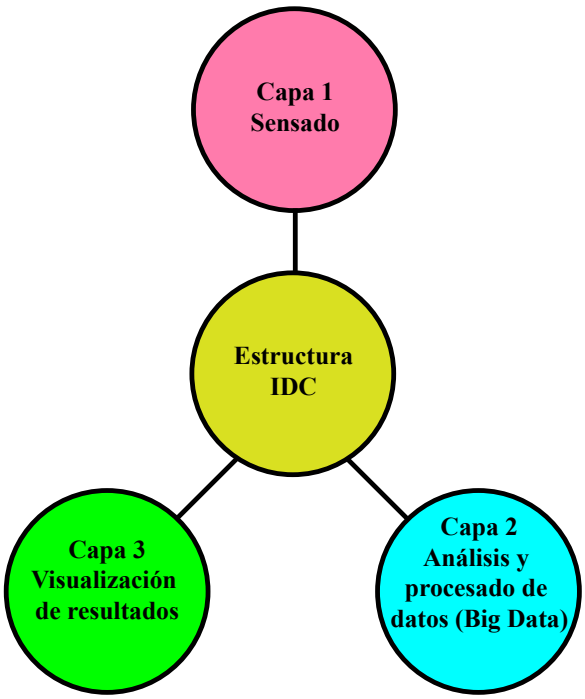

Figura 4.1: Esquema de IDC

4.3.1. Capa 1: Sensado

La capa de "sensado", recopila del ambiente muestras o mediciones realizadas en tiempos establecidos. En la misma capa se considera la ejecución de las tareas usando actuadores. Dependiendo de la complejidad de la tarea a desarrollar, en esta capa se pueden tomar decisiones y ejecutar acciones; un ejemplo de ello se tiene en líneas de producción en donde el intervalo entre mediciones es de poca latencia, es decir, que dada la medición la respuesta de ejecución es inmediata. Para estos casos se le informa al IDC, la presencia del evento y su registrado, sin embargo, la toma de decisión ejecutada se desarrolla en el propio ambiente. Por lo tanto, la capa de sensado, para algunos casos, contiene un sistema de procesamiento de bajo nivel y contiene algunas tareas de ejecución establecidas.

En ella se tiene, además, el proceso de conversión de las mediciones en forma analógica a su generación digital y el cumplimiento de los protocolos de transmisión para lograr la conexión con la plataforma de internet. Para realizar la comunicación vía el internet, se emplea el protocolo *IPv6*, que hace referencia a *Internet Protocol version 6*. En el internet de las cosas, por tanto, se manejan datos en un esquema bi-direccional y de tipo cifrado para ofrecer seguridad en la transmisión [65].

El monitoreo constante del ambiente puede ser realizado por varios dispositivos a la vez; con ello, es posible confirmar o ratificar estados previamente programados. Un problema que se presenta es la cantidad de cableado para lograr las conexiones entre diferentes dispositivos; para evitar este problema se han generado protocolos de trasmisión inalámbrica, de los más populares podemos citar los protocolo Bluetooth y Wifi. Por la búsqueda de interconexión con otros dispositivos y considerando que la plataforma de internet es internacional, esta capa contiene protocolos que son normados para poder trabajar en cualquier parte del mundo.

Para el internet de las cosas, se considera que esta capa contiene la parte de ingeniería dura, es decir, la parte física en la ejecución de la toma de decisiones y la electrónica requerida por el sistema.

4.3.2. Capa 2: Análisis y Procesamiento de los Datos

Esta capa inicia con la estructuración de los datos, es decir, los datos son ordenados considerando diferentes directivas. Una de estas directivas, es la de medición; otra de ellas es el identificador de qué dispositivo realizó la medición, fecha y registro de la propia medición. A este proceso se le nombra *Estructuración de la Base de Datos*. El almacenamiento de los datos

ya estructurados, facilita el análisis de los datos, ya que al llevar el histórico de forma ordenada se pueden encontrar patrones de comportamiento del ambiente a controlar y al interactuar con seres humanos, llegar a conocer sus preferencias. El inconveniente primordial en esta capa, es el almacenamiento de grandes cantidades de información. Una magnitud de medición, comúnmente empleada es el Gigabyte (10^9 *bytes*). Aunque en los últimos años, se ha vuelto común hablar en términos de Terabytes (10^{12} *bytes*) de información, o también miles de Gigabytes.

Para el procesamiento de toda esta información se han tenido que desarrollar nuevas técnicas y muchas de ellas consideran el procesado en paralelo de los datos. Para este trabajo se han implementado dos técnicas emergentes: la primera de ellas, son los algoritmos metaheurísticos inspirados en el comportamiento organizacional de las abejas; la segunda, es la incursión de redes neuronales artificiales, biológicamente plausibles, conocidas como redes con neuronas tipo spiking, para la reducción en la dimensionalidad de los datos.

4.3.3. Capa 3: Visualización de los Datos

En esta capa el usuario logra interactuar con el sistema, en términos de computación se le llama *"interfaz de usuario"*. El usuario final, en la mayoría de las ocasiones, solo percibe la ejecución o toma de decisión del sistema; si consideramos que el sistema controla la temperatura de una habitación, el usuario final, únicamente observa el valor numérico de la temperatura.

Existe un segundo usuario, conocido como *"intermedio"* que en sí es el programador del sistema y es quien gestiona, empleando la computadora, la parte del procesamiento de los datos y el cálculo para la toma de decisiones. En esta capa el usuario intermedio, tiene accesos a las bases de datos y a los algoritmos que se emplean para ejecutar las tareas. Es importante mencionar que en esta misma capa se actualizan los datos y en su momento se pueden modificar los patrones establecidos previamente. Una parte importante en el internet de las cosas es la capacidad de adaptabilidad ante el registro de nuevos datos que pueden indicar modificaciones en el comportamiento del ambiente o medio de interés. Continuando con el ejemplo de la temperatura de una habitación, esta temperatura podría variar en un solo día y a su vez según la época o estación del año en que se encuentre.

La interacción de las tres capas da nombre al propio concepto del internet de las cosas. A mediados del 2010, el número estimado de objetos conectados a internet fue de alrededor de 12,5 mil millones y para el año 2015 se estimó en 25 mil millones, con lo que el número de dispositivos inteligentes se duplica en poco tiempo. En el 2020, el cálculo se estimó en otros 50

mil millones [66,67]. Sin lugar a dudas, el paradigma del internet de las cosas bien representa el concepto de ambientes y lugares inteligentes.

4.4. Conclusiones del Capítulo

El paradigma del internet de las cosas y sus componentes han sido revisados en este capítulo. Dentro de sus principales virtudes se puede mencionar la adaptabilidad a diferentes ambientes de aplicación, sin embargo, su desempeño depende de la calidad de los datos recabados por los sensores que integran el sistema.

Con lo revisado en este capítulo, es esencial considerar la calidad de la conexión al internet, ya que para algunos procesos el tiempo que transcurre entre el registro de un dato y su análisis para la toma de una decisión, juega un papel importante.

Se consideraron tres ambientes de aplicación del internet de las cosas: Ciudad, Cultivo y Hogar inteligente. Los ambientes considerados son distintos en cuanto a sus parámetros de medición, sin embargo, los datos generados son almacenados y analizados de manera similar.

5

Base de Datos

Con la aparición del paradigma del internet de las cosas, cada día el volumen o cantidad de datos a procesar se incrementa, sin olvidar que día con día se suman nuevos ámbitos donde se requiere el procesamiento de información. Podemos citar que las empresas comerciales, necesitan procesar información para dirigir sus productos en un mercado específico; en las ciudades o comunidades, las autoridades quieren optimizar el consumo de sus recursos, y las personas en casa buscan una mayor comodidad.

El dilema principal no es la captura y el almacenamiento de los datos, sino la interpretación de ellos y los patrones que se extraen para la toma de decisiones. Todos estos beneficios se logran con el procesado de bases de datos y han dado origen a la llamada minería o interpretación de datos; para esta tarea, es necesario trabajar con un conjunto de datos y en su momento combinarla con otros conjuntos de datos, lo que hace que el tamaño de la información aumente de forma geométrica o exponencial. Se tiene entonces: *"Volumen de información"*, asociada a la: *"Variabilidad de los datos"* y que en conjunto también conduce a la: *"Validez en el Tiempo"*, ya que pueden cambiar de forma instantánea, por lo mismo, se tiene en consideración la *"Velocidad de crecimiento de los datos"*, por lo que muchos expertos en el tema, han definido que se tiene que lidiar con los cuatro tipos de V.

Por lo antes mencionado, se tiene la dificultad en la captura, el manejo y procesamiento de los datos con técnicas convencionales, por lo que es necesario explorar nuevos algoritmos y técnicas para el procesado fuera de lo convencional.

5.1. Definición de Big-Data

Hasta el momento, no hay una definición concreta y aceptada del término *Big Data*; sin embargo, en este trabajo presentaremos la definición dada por McKinsey Global Institute [68], que un informe en mayo del año 2011 citó: *"Big-Data se refiere a los conjuntos de datos cuyo tamaño está más allá de las capacidades de las herramientas típicas de software de base de datos para capturar, almacenar, gestionar y analizar"*. Está definición no aborda la subjetividad al determinar qué tan grande necesita ser un conjunto de datos para ser considerado como Big Data; sin embargo, es frecuente asociar el término de Big-Data a Gigabytes, Terabytes o Petabytes de información.

Con la minería de datos y las bases de datos actuales surge la pregunta ¿Por qué el Big Data es importante? Una posible respuesta es que, ayuda a dar respuestas sobre comportamiento de eventos, tendencias sobre las elecciones y preferencia sobre un grupo de personas de manera local, regional e incluso internacional, y sobre todo, con estos análisis poder realizar predicciones.

Con Big Data se puede incluso plantear preguntas y respuestas de eventos que aún no suceden; un ejemplo de esto, se encuentra actualmente en la búsqueda de una dirección empleando el celular, que además de proporcionar la dirección solicitada, proporciona, dependiendo de la hora del día, sugerencias adicionales, como por ejemplo: lugares para comprar, comer, divertirse, etc., considerando los hábitos conocidos del usuario.

De manera actual, se tienen sistemas que además de capturar datos, pueden generar datos similares pero ficticios, por lo que una tarea que actualmente se realiza es comprobar la veracidad de los datos, ya que muchos de ellos son redundantes y contienen rastros de duplicidad. Una manera de dar solución a estos problemas es la reducción de dimensionalidad de los datos, ya que es posible determinar si una característica aporta información relevante al analizar grandes cantidades de datos.

Inicialmente, las empresas contaban con los datos recabados y estructurados por ellos, sin embargo, actualmente es un error considerar únicamente datos propios, por lo que es una práctica común conjuntarlos con los datos que se obtienen de redes sociales, para tener una visión externa de la propia empresa, su producto y su aceptación. El inconveniente de tomar bases de datos externas es que pueden o no contener una estructura establecida para esos datos y quizás ser no compatibles con las bases de datos propias; a este problema se le nombra “bases de datos de estructuras complejas”, por lo que los nuevos algoritmos deben ser flexibles en poder procesar

diferentes bases de datos y adaptarse a diferentes estructuras o formas en que se han capturado los datos [69].

Las fuentes más comunes para recopilación de datos son:

- Datos de internet y móviles.
- Datos extraídos con el internet de las cosas.
- Datos recopilados por encuestas.
- Datos experimentales o simulaciones de eventos.
- Datos extraídos de redes sociales (Facebook, Twitter, etc.).

Los tipos de base de datos o estructuras de almacenamiento son:

- Bases de datos no estructurados (datos en desorden, sin ningún lineamiento en su almacenamiento, audios sin registros, segmentos de video, listas de usuarios sin registros).
- Bases de datos semi-estructurados(datos con un registro único, no global, hojas de cálculo, reportes, uso de software).
- Bases de datos estructuradas (bases de datos globales con diferentes lineamientos ordenados para su almacenamiento).

Para este trabajo se ha propuesto analizar tres bases de datos que por su tamaño pueden ser consideradas del tipo Big-Data y que por la diversidad de las fuentes de información se demuestra la adaptabilidad del sistema propuesto en este trabajo. Estas bases de datos son descritas a continuación.

5.2. Base de Datos de Números Escritos a Mano

En los últimos años un tema que ha tomado auge es la seguridad electrónica y se han propuesto diferentes soluciones, así se han desarrollado sistemas para detección de retina, huellas dactilares, la grafología, la morfología en la escritura de los números, etc. En sistemas de seguridad es importante tener la mayor información posible y que represente el menor esfuerzo en los usuarios, por lo que una medida de seguridad es pedirle al usuario que recuerde un código de seguridad numérica; este código puede ser escrito en una pantalla táctil por el usuario, extrayendo de este proceso la proximidad relativa de los números así como la forma de ellos.

La primera base de datos en esta sección, se refiere a la escritura de números de un grupo de 44 personas. Se ha considerado, como primer conjunto de características, las posiciones que

ocupa en píxeles, la escritura de los números correspondientes del 0 al 9. Para esta tarea se ha considerado una Tableta de 500×500 píxeles. También, se cuenta con un sensor de presión que ayuda a extraer el nivel de presión ejercida en el momento de la escritura de los números, completando para cada evento ejecutado un conjunto de 16 atributos [70].

5.3. Base de Datos de Flores, Frutas y Rostros

La segunda base de datos está constituida por un conjunto de imágenes de flores, frutas y rostros [71]. Las imágenes de flores, contienen diferentes clases de ellas y sus principales características son la forma en sus pétalos y la textura que presentan. Para las frutas se tienen como principales características la textura y sus estructuras que, en la relación a las dimensiones de las imágenes, son más grandes en comparación a las dimensiones de los posibles pétalos en las flores. Para los rostros, se considera la estructura y la presencia de características constantes y de baja variación, es decir, en ellas se tendrán patrones de ojos, boca, nariz, y cejas. Sin embargo, comparten características entre ellas que hacen complejo determinar cada clase de objetos. Por lo tanto, es necesario determinar y extraer las características determinantes que definen cada tipo de objeto y eliminar las características redundantes y que provocan confusión en la clasificación.

Para cada tipo, se ha considerado la cantidad de 100 imágenes. Las imágenes se encuentran expresadas en escala de grises, de donde el tono más obscuro (negro) es asociado al valor numérico 0 y el tono más claro (blanco) al valor numérico 255, con una dimensión de 100 filas por 100 columnas. De cada imagen se han extraído 67 características, que corresponden a los coeficientes entregados por la técnica de procesamiento de imágenes llamada *Patrón Binario Local* (LBP); esta técnica ha sido considerada, ya que se reporta un buen desempeño para la extracción de características de textura, estructura y reconocimiento facial [72–74]. Para su implementación se emplea un barrido de bloques de 64×64 píxeles, por toda la imagen, de donde, para cada bloque se obtiene un histograma relativo a las variaciones de los niveles de grises en esa área. El vector de reconocimiento para cada imagen está formado por los diferentes histogramas que se obtienen del conjunto de bloques que se requierán para cubrir toda una imagen (las dimensiones del bloque dependerán del tamaño de la imagen a procesar).

5.4. Base de Datos de Terrenos de Cultivo

La incursión de nuevas tecnologías en la agricultura es un área en crecimiento, muchos países han apostado por tecnificar los cultivos buscando mejorar la producción de alimentos. Para ello se tienen redes de sensores en el terreno que dan información sobre la calidad del suelo, pero para determinar la presencia de plagas y/o daños en los cultivos es común recurrir a imágenes aéreas, por lo que se construyen naves no tripuladas (drones), con sensores ópticos que almacenan imágenes a diferentes longitudes de onda; con ellos se puede determinar la presencia de plagas, enfermedad en las plantas y el desarrollo del cultivo.

La tercera base de datos considerada es referente a terrenos de cultivo; se consideran atributos extraídos de un conjunto de sensores colocados en una nave no tripulada, que ejecuta rutas de vuelo sobre terrenos de cultivo cerca de Winnipeg, Manitoba, Canadá. El sensor principal es un radar óptico bitemporal fusionado para la clasificación de tierras de cultivo [75], además, de contar con una cámara de tipo polarimétrica. Las muestras tomadas por el radar son en dos tiempos distintos con 49 valores en cada uno de ellos. De una manera similar, la cámara realiza dos tomas de muestras de 38 datos cada una. Por lo tanto, cada muestra se compone de 174 características. Se han mapeado sietes zonas de cultivos correspondientes a: 1 -Maíz; 2 - Guisantes; 3 -Canola; 4 - Soya; 5 - Avena; 6 - Trigo; y 7 - Cultivos de hoja ancha.

La Tabla 5.1 contiene un resumen de las tres bases de datos consideradas para este trabajo. Las tres bases de datos son variadas en cantidad de características, además de presentar diferentes ámbitos de medición. En todas ellas, se ha logrado la reducción de dimensionalidad, manteniendo las condiciones de distribución de los datos en su forma original y los datos obtenidos son presentados en el capítulo 7.

Tabla 5.1: Parámetros de las Bases de Datos

Nombre	**Datos**	**Características**	**Clases**	**Dimensión**
Escritura de Números	1000	16	10	1000×16
Frutas, Flores, y Rostros	300	67	3	300×67
Tierras de Cultivo	700	174	7	700×174

5.5. Conclusiones del Capítulo

En este capítulo se han presentado los aspectos más importantes a considerar en el diseño y manejo de una base de datos. Para una base de datos se deben de considerar cuatro aspectos

importantes: el volumen de la información almacenada, la variabilidad de los datos, el tiempo en que los datos son válidos o de interés, y la velocidad con la que crecen los datos capturados. Si en uno de estos aspectos se falla, el sistema es considerado como vulnerable o de baja eficiencia.

El paradigma conocido como Big-Data ha sido definido y se ha planteado la normatividad a seguir para estructurar y almacenar la información. La estructura con la que es almacenada la información, determina la flexibilidad de la propia base de datos. Por lo tanto, es importante la implementación de una base de datos flexible, lo cual permite tener acceso a la información en más de una manera, lográndose así que los datos puedan ser analizados desde diferentes directrices.

6 Reducción de Dimensionalidad

6.1. Introducción a la Reducción de la Dimensionalidad de los Datos

Uno de los temas de investigación del aprendizaje en máquina (o también llamado Machine Learning) está relacionado con la reducción de la dimensionalidad de datos complejos, es decir, descubrir las clases y la información relevante que las representa.

En el análisis de datos (minería de datos), un problema común es trabajar con datos que están descritos por múltiples variables o características, que hacen que el conjunto de datos sea enorme en información, complicando su análisis y aumentando el coste computacional en su tratamiento. Por lo tanto, un objetivo a lograr con la reducción de la dimensionalidad es transformar un conjunto de datos en un espacio de alta dimensionalidad a un espacio de baja dimensionalidad, pero conservando la distribución original de los datos.

Es pertinente el responder el por qué es necesario el proceso de la reducción de los espacios de información antes de realizar el análisis de los datos y generar la toma de decisiones o la obtención de modelos de predicción. Una primera respuesta es visualizar los datos con la finalidad de conocer y comprender el problema que se esté procesando. Una segunda respuesta sería que la reducción de dimensionalidad servirá a procesos posteriores, donde se aplican modelos de reconocimiento de patrones o agrupamiento de datos, logrando disminuir los tiempos de procesamiento y generación de decisiones. Por lo tanto, los algoritmos de reducción de dimensionalidad deben poder ser descritos en términos de tres conceptos:

- Reducir el espacio de dimensión de los datos, manteniendo la distribución original de los mismos.
- Reducir el tiempo y consumo de recursos computacionales para subsecuentes procesos.

- Lograr generar espacios de menor dimensión donde sea posible visualizar la distribución de los datos.

Como siguiente paso, se considera la evaluación del resultado de la reducción del espacio dimensional. Para esta tarea se ha considerado aplicar el cálculo de la matriz de co-ranking, que implica comparar la distribución original de los datos contra la distribución obtenida del proceso de reducción de dimensión; debido a su baja complejidad matemática y fácil implementación, es un método muy empleado para determinar la calidad de la reducción de dimensión en los datos. La descripción de la metodología del cálculo de matriz de co-ranking, se incluye como un subtema dentro de este capítulo.

En la temática de la reducción de dimensionalidad se tienen principalmente modelos matemáticos lineales y no lineales.

En el sentido lineal de reducir la dimensionalidad, se ha aplicado el método clásico de Análisis de Componentes Principales (PCA), el cual calcula los vectores y valores propios de la matriz de covarianza del conjunto de datos. Estos vectores y valores propios indican nuevas direcciones que son consideradas direcciones principales o donde la variación de los datos es menor y esto corresponde a versiones reducidas de la información contenida en los datos.

En el campo de las redes neuronales, se han propuesto modelos basados en procesos lineales; un ejemplo es la arquitectura nombrada *"Autoencoder"*, la cual desarrolla la extracción de la información más relevante de una base de datos, y su proceso de entrenamiento es equivalente al desarrollado por el método de PCA.

Dentro de los métodos no lineales, podemos nombrar la técnica de t-SNE, que se basa primeramente en determinar la distribución de probabilidad de los datos en alta dimensión; como segundo paso, crea un modelo de baja dimensión con la normativa de conservar la distribución de probabilidad inicial.

En este trabajo se ha considerado realizar una variante del modelo t-SNE; se ha propuesto incluir una red neuronal de tipo Spiking, que se encarga de generar la reducción de dimensión de los datos iniciales. Esta técnica, considerada de tipo no lineal, genera dos distribuciones de probabilidad, la primera es para los datos de entrada y la segunda para la propuesta de reducción de dimensión a la salida.

Los sistemas de autoencoder y t-SNE son descritos en las siguientes secciones.

6.2. Autoencoder

La arquitectura neuronal de tipo autoencoder [76] es presentada en la Figura 6.1. Esta arquitectura cuenta con dos etapas principales, la primera de ellas nombrada como encoder, o codificador, se encarga de cifrar los datos en una representación reducida de ellos. La segunda etapa, nombrada como decoder o decodificador, se encarga de tomar la representación reducida en el proceso anterior y realizar la expansión para obtener nuevamente la representación de los datos de entrada. Parece ser un proceso con poco sentido, ya que los datos a la entrada de la arquitectura neuronal son nuevamente recuperados a la salida; sin embargo, la tarea más importante es la que se lleva a cabo en la parte intermedia, ahí se logra obtener una versión reducida en dimensión de los datos originales.

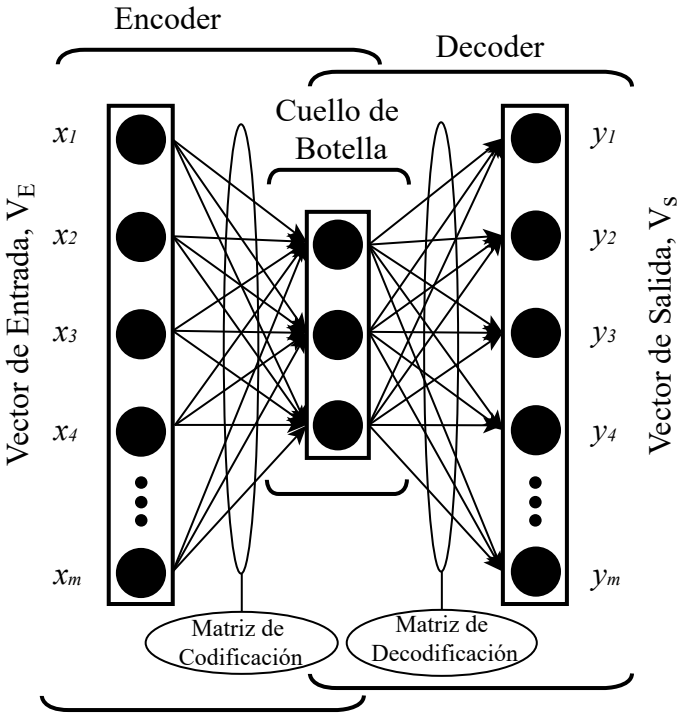

Figura 6.1: Arquitectura Neuronal Autoencoder

Para la parte del *"Encoder"*, se tiene el vector de entrada (V_E), que es construido con todas las variables o características que se emplean para representar cada uno de los datos; consideramos el vector de datos de entrada $X = [x_1, x_2, ..., x_m]$ con m características.

Este vector de entrada es presentado a la arquitectura y es multiplicado por la matriz sináptica; para el caso presentado se tienen 3 neuronas a la salida con m características, en este punto la matriz sináptica es considerada como el *vector de ponderación de codificación* y la salida es el arreglo neuronal que contiene la versión de características reducidas de los datos. Para

determinar el proceso de entrenamiento en la sección del encoder, se emplea la Ecuación (6.1), el resultado de la ecuación es representado por el vector $\vec{a}$ que se obtiene al aplicar una función de transferencia lineal al producto de la matriz de codificación, $W_{3,m}$, por el vector de entrada, $\vec{V}_E$, sumando los valores de polarización de las neuronas (ver la sección 3.2). La arquitectura que se emplea para el encoder es presentada en la Figura 6.2.

$$\vec{a} = f_{lineal}\left(\left(W_{3,m} \cdot \vec{V}_E\right) + \vec{b}_{encoder}\right) \tag{6.1}$$

El vector que se obtiene a la salida del enconder, se considera como la entrada de la decodificación. Este vector es considerado como el cuello de botella, ya que en él se compacta la información y los vectores iniciales se ven reducidos en sus dimensiones.

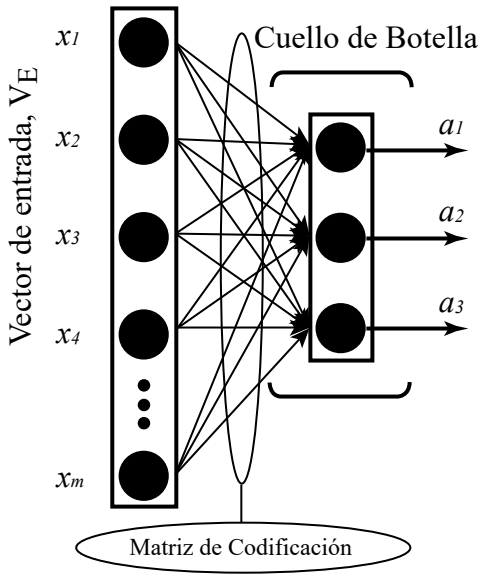

Figura 6.2: Arquitectura Neuronal Correspondiente al Encoder

El *"decodificador"* se encarga de expandir y reconstruir el vector original considerando que en la entrada el vector de características se ha comprimido. El proceso de la decodificación asegura que los vectores compactados contienen la información relevante y necesaria para representar los datos en un espacio de características reducido. Esto se comprueba, presentando un vector en la entrada y verificando que el vector de salida (V_s) es similar al vector de entrada del autoencoder. El vector de salida V_s es representado por $Y = [y_1, y_2, ..., y_m]$, y como es de esperar, contiene la misma cantidad de características m que el vector de entrada.

El proceso para realizar la decodificación es calculado empleando la Ecuación (6.2) donde se considera la aplicación de una función lineal, f_{lineal}, al producto del vector de ponderación, $W_{m,3}$,

en la decodificación, por el vector de salida del encoder, a, sumándole el vector de polarización, $\vec{b}_{decoder}$, para cada neurona. La arquitectura empleada para el decodificadore es presentada en la Figura 6.3.

$$\vec{V}_s = f_{lineal}\left((W_{m,3} \cdot \vec{a}) + \vec{b}_{decoder}\right) \tag{6.2}$$

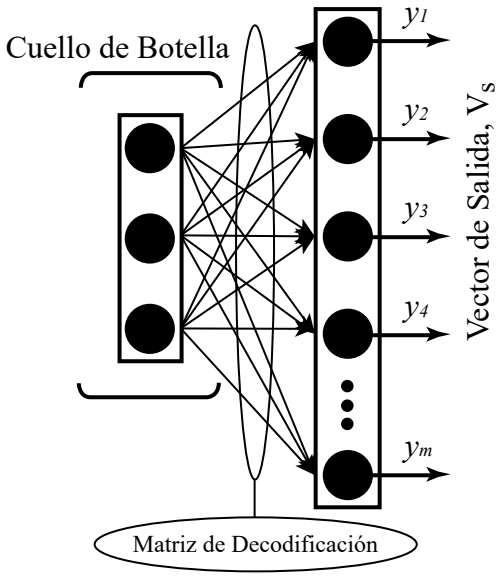

Figura 6.3: Arquitectura Neuronal Correspondiente al Decoder

Para el entrenamiento de la arquitectura neuronal del Autoencoder, se emplea el método de aprendizaje *"BackPropagation"*(ver sección 3.2), de donde, para el proceso inicial de *"FeedForward"* se presenta a la arquitectura cada uno de los vectores de entrada para obtener los valores del autoencoder. El vector de salida del autoenconder es comparado con el vector de referencia *"target"*, para obtener un valor de error y aplicar el proceso de propagación del error en retroceso (los vectores de datos en la entrada son considerados como los vectores de referencia).

En el proceso de entrenamiento de la arquitectura neuronal Autoencoder, se realiza el cálculo del "Error Cuadrático Medio (ECM)", entre los valores de entrada y de salida. Cuando la magnitud de ECM es cercana a cero, se puede considerar que la extracción o reducción de la dimensión de los datos en la capa intermedia, es pertinente. Esto debido a que se ha logrado, con los datos esenciales extraídos en la capa intermedia (o cuello de botella), la reconstrucción en la capa de salida, de los datos iniciales.

6.3. t-SNE

El método conocido como: *t-SNE(t-distributed Stochastic Neighbor Embedding)* [77], se propuso como un método alternativo para visualizar información compleja en un espacio con dimensiones reducidas, pero optimizando la calidad del proceso de agrupación para determinar agrupaciones en colecciones de datos. Para esta tarea, primero se analizan los datos de entrada y de ellos se generan agrupaciones que contengan algún rasgo en común. Cada muestra, del grupo de datos, es comparada con el resto de las muestras obteniendo su medida de distancia para determinar los vecinos cercanos, generando así, agrupaciones. Para el método t-SNE un parámetro a determinar es el de "Perplejidad" y representa la cantidad de la muestra que se desea considerar como parte del vecindario.

Cada muestra del grupo de datos de entrada, es considerada como el valor medio de una distribución de probabilidad gaussiana y el valor de desviación estándar representa el radio de acción o vecindario. Las distancias calculadas son empleadas para determinar los valores de probabilidad, desarrollando la Ecuación (6.3).

$$p_{ij} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\left\|x_k - x_l\right\|^2 / 2\sigma^2\right)} \tag{6.3}$$

La similitud entre la muestra x_i y la muestra x_j es la probabilidad condicional $p_{i,j}$ de que x_i elija a x_j como vecino, considerando que los vecinos son elegidos en proporción a la densidad de probabilidad bajo una campana gaussiana centrada en x_i y con una varianza σ^2. En términos simples, cada muestra o punto de la base de datos es considerado como el centro de una distribución de probabilidad y en el vecindario se encuentran los puntos más afines a él. Con este cálculo, se tiene la ventaja de poder determinar varianzas locales, (contrario al Autoencoder que obtiene varianzas globales), por lo que se pueden realizar agrupaciones pequeñas de la misma clase; posteriormente se integran las clases similares hasta formar agrupaciones mayores. Así es posible analizar bases de datos con distribuciones no lineales, como la que se muestra en la Figura 6.4. Para este tipo de distribuciones, los métodos lineales no son factibles de aplicar para la tarea de agrupamiento de datos.

En la Figura 6.4, se presenta una base de datos de prueba (Landmark) comúnmente empleada para evaluar el desempeño de sistemas de reducción de dimensionalidad [78]. En la Figura 6.4(a), se presenta una vista isométrica a 21° grados y se representan 4 clases, en donde cada clase se

representa con un color. En la Figura 6.4(b), se presenta la vista del plano $Y - Z$, en donde se observa la mezcla de los datos para todas las clases y es posible deducir la complejidad de la tarea de la separabilidad de los datos. Finalmente, en la Figura 6.4(c) se muestra la respuesta del método t-SNE y se puede observar que el método genera agrupaciones correspondientes a cada clase, identificadas por un color.

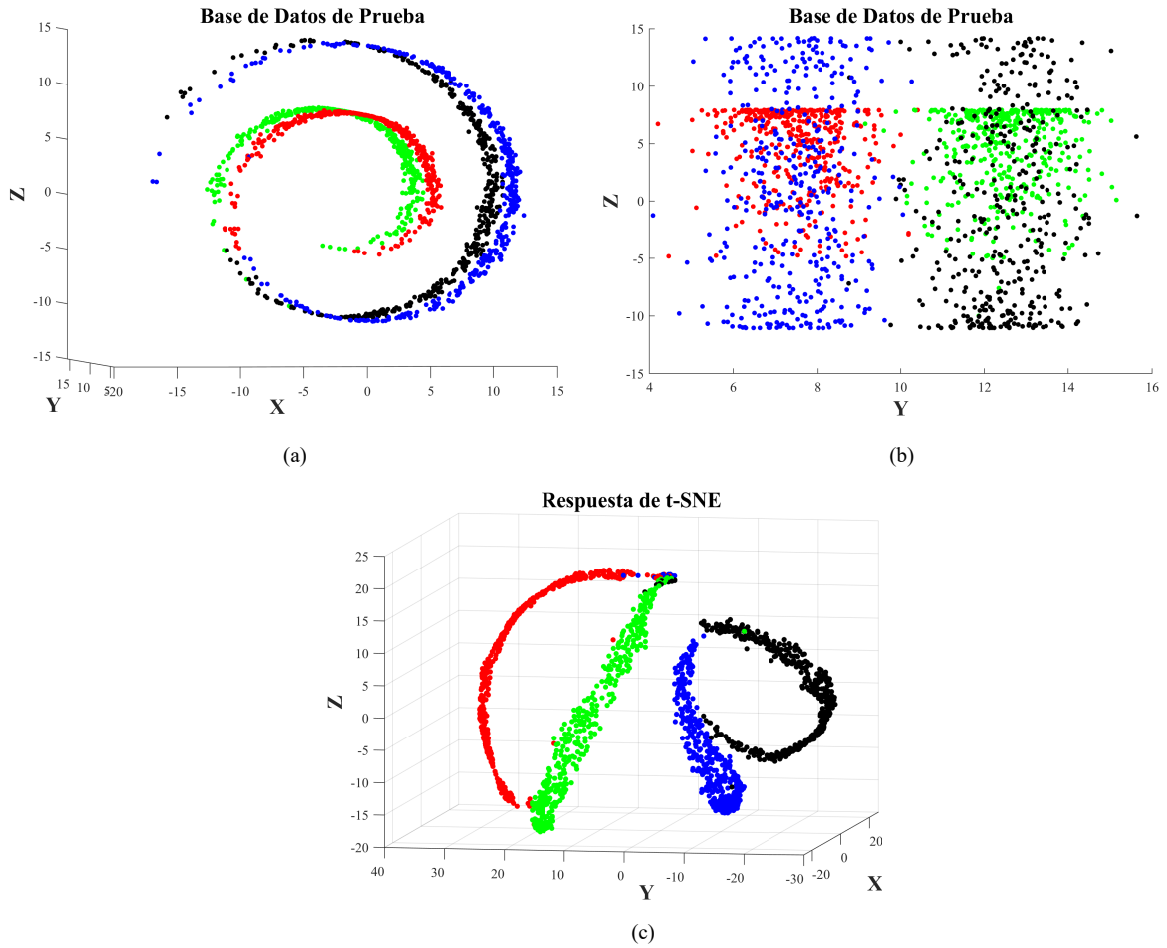

Figura 6.4: Base de datos de Prueba. (a)Vista Isométrica a 21° de los datos iniciales. (b)Vista del plano $Y - Z$ de los datos iniciales. (c)Respuesta del método t-SNE

En el método de t-SNE, la distribución de salida tiene la misma cantidad de muestras que en la base de datos original, pero con una menor cantidad de características; generalmente se proponen 2 o 3 dimensiones, con la finalidad de poder observar la distribución de los datos. Un proceso similar, al del cálculo de funciones de densidad de probabilidad, es desarrollado con los datos propuestos a la salida, pero con la diferencia de que la *función de densidad de probabilidad es de Cauchy* y con un grado de libertad; esta función de distribución es la de *t-student*. La distribución t-student se aplica a los datos de salida debido a que se consideran valores de probabilidad más altos y cercanos a la media, logrando de esta manera trabajar con

distribuciones de datos más compactas y con menores características para su representación. La Ecuación (6.4) es aplicada para obtener la función de distribución de probabilidad de los nuevos datos.

$$q_{ij} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq l} \exp\left(-\left\|y_k - y_l\right\|^2\right)} \tag{6.4}$$

De la Ecuación (6.4), se obtienen las funciones de probabilidad $q_{i,j}$, que representan la probabilidad de que la muestra y_j, sea elegida como parte del vecindario de la muestra y_i y se continúa desarrollando este proceso sobre todas las muestras de salida y_i, que como hemos mencionado son las versiones reducidas de las muestras iniciales x_i. Los valores de y_k y y_l, representan todas las combinaciones posibles de los términos y_i y y_j.

Si los puntos propuestos y_i mapean la distribución de los puntos de alta dimensión x_i, las funciones de probabilidad condicional en p_{ij} y q_{ij} deberán ser iguales. Ahora bien, para determinar que los datos reducidos propuestos mantienen la distribución de los datos originales, se desarrolla el cálculo de semejanza entre ambas distribuciones de probabilidad, calculando el valor de la divergencia de *Kullback-Leibler (K-L)*. Para el caso genérico, se puede decir que es la entropía cruzada de la sumatoria de las probabilidades en p_{ij} y q_{ij}. Esta divergencia se encuentra representada en la Ecuación (6.5).

$$C = KL\left(P||Q\right) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{6.5}$$

Partiendo de la idea de una representación de datos con baja dimensión que minimice la diferencia entre ambas funciones de distribución de probabilidad, se busca minimizar la magnitud para la divergencia *K-L*. Por lo tanto, el problema de la generación de datos de baja dimensión que mapea a la distribución original, puede verse como un problema de optimización, de donde se busca determinar los valores en los parámetros que hagan mínima a la divergencia *K-L*. En este trabajo se propone resolver este problema de optimización empleando el algoritmo *ABC (Artificial Bee Colony)* y para la generación de las muestras de salida y_i, una arquitectura neuronal basada en neuronas spiking.

6.4. Arquitectura de Red Neuronal Spiking de Tipo No Lineal

En la sección 3.3, se presentó la neurona spiking con respuesta sigmoidal y se le nombró neurona spiking de respuesta no lineal. Empleando estas neuronas, se propone una arquitectura de red neuronal con la capacidad de procesar las muestras de una base de datos y generar una colección de nuevas muestras de dimensiones reducidas para ser consideradas como la distribución con reducción de dimensión. La red neuronal propuesta se observa en la Figura 6.5.

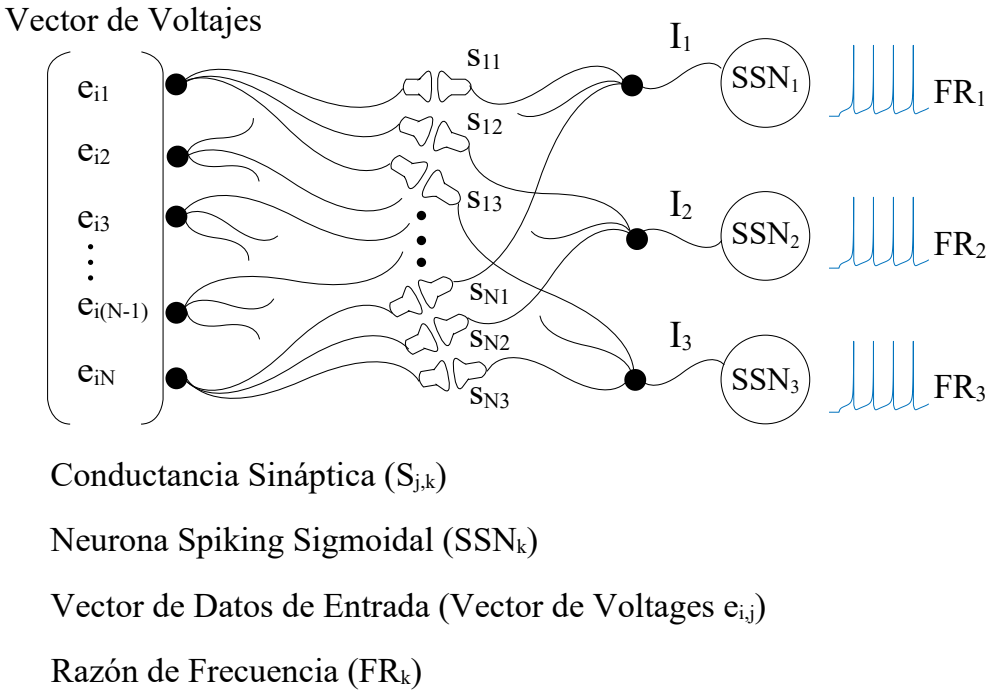

Figura 6.5: Arquitectura de la RNS Propuesta

La arquitectura está formada por 3 neuronas: SSN_1, SSN_2 and SSN_3; cada una y de manera independiente, generan pulsos por unidad de tiempo y de ello se obtiene el valor de frecuencia etiquetados como: FR_1, FR_2 and FR_3, respectivamente y conforman los valores de las características de las nuevas muestras reducidas, que para este estudio de doctorado corresponde a un espacio de 3 dimensiones. Estas neuronas reciben las corrientes I_1, I_2 y I_3, que son obtenidas al multiplicar los datos de entrada con los valores de los pesos sinápticos. Los valores numéricos de las muestras originales en la entrada son considerados como valores de voltajes $\{e_{i,j}\}$ que, al ser multiplicados por los valores del conjunto de conductancias o pesos sinapticos $\{s_{j,k}\}$, generan los valores de corriente que son suministrados a cada neurona.

Si las conductancias (pesos sinápticos) son modificadas se obtendrán valores diferentes en la razón de frecuencia de pulsos generados por las neuronas spiking; esto representa generar una

nueva distribución en los datos a la salida. Por lo tanto, es posible ajustar los valores de las conductancias para lograr que la distribución de salida sea semejante a la distribución de la entrada, lo cual representa que la divergencia *Kullback-Leibler*, se minimice. Las conductancias de la arquitectura neuronal, representan los parámetros a ser ajustados por el algoritmo de Abejas y la divergencia K-L, se emplea como la función a ser optimizada.

6.5. Método t-SNE Spiking Neuronal

Con lo descrito en las secciones anteriores, en este capítulo es posible modificar el modelo de reducción dimensional t-SNE, combinándolo con la arquitectura neuronal spiking no lineal. En el diagrama a bloques de la Figura 6.6, se presenta la propuesta del método t-SNE spiking neuronal.

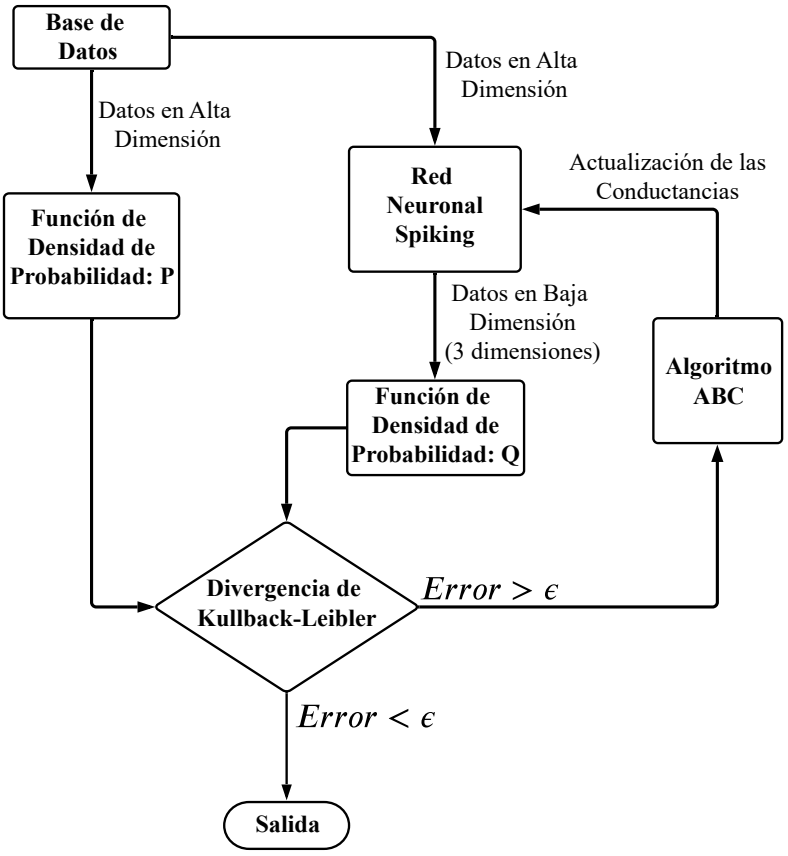

Figura 6.6: Diagrama de Flujo del Método Propuesto en este Trabajo de Doctorado t-SNE Spiking Neuronal

Siguiendo el diagrama de flujo, el bloque *Base de Datos* representa la entrada de la base de datos estructurada con alta dimensión. Este bloque alimenta tanto al bloque encargado de obtener la función de densidad de probabilidad, como al bloque que constituye la red neuronal

spiking. El bloque que calcula la *función de densidad de probabilidad* P produce el cálculo considerando a los datos sin reducción.

El bloque *Red Neuronal Spiking* contiene la arquitectura neuronal empleada, en donde cada bloque de neurona spiking es de tipo no lineal. Es importante resaltar que la tarea principal de este bloque es la generación de los datos, pero con una disminución en la representación de sus características; para el caso particular de este trabajo se ha realizado una visualización a tan solo tres características.

La propuesta de los datos con una reducción en sus dimensiones, alimenta al bloque encargado de procesar y generar la *función de densidad de probabilidad* Q que, en conjunto con P, alimentan al bloque llamado *Kullback-Leibler Divergence*. Este bloque determina qué disparidad existe entre las dos funciones de probabilidad y otorga como información el valor numérico de similitud, el cual se toma como un valor de error.

Si el valor de error, determinado para la distribución actual Q, no ha alcanzado un umbral deseado, se le informa al bloque que representa la ejecución del algoritmo de abejas ABC. Este bloque modifica los valores de los pesos sinápticos neuronales, provocando con ello que la arquitectura neuronal responda con valores diferentes para cada dimensión o características que se hayan propuesto; de esta manera se logra modificar la representación de los datos de baja dimensión. Los pesos sinápticos representan los valores de conductancias de las neuronas, los cuales al ser modificados, las neuronas cambian la razón de frecuencia en la generación de los pulsos o spikes de respuesta. La Ecuación, que calcula la divergencia de Kullback-Leibler, se convierte en la función a optimizar en los procesos de ajustes del algoritmo de abejas, ABC. Una vez alcanzado el valor de error mínimo establecido, el proceso es detenido.

6.6. Análisis del procesamiento t-SNE

La reducción de dimensionalidad es una tarea que en los últimos años ha tomado relevancia y por lo tanto se han reportado diversas técnicas para su realización. En este capítulo, en la su segunda sección, se ha abordado la reducción de dimensionalidad desde un enfoque de procesos matemáticos lineales; si bien se presenta una arquitectura con neuronas artificiales, el sistema de manera generalizada logra realizar el proceso de reducción, pero únicamente sobre datos que mantienen una relación lineal.

Las arquitecturas neuronales de codificación y decodificación son ensambladas para obtener

un sistema con la capacidad de procesar los datos y extraer sus componentes principales. Como método de prueba se reconstruyen los datos a partir del decodificador, con la idea de comprobar que la extracción de las componentes principales es la adecuada. Sin embargo, se mantiene la limitante de trabajar con datos con comportamiento lineal.

Como alternativa en el proceso de reducción de dimensión de los datos de tipos lineal y no lineal, se ha considerado la técnica t-SNE, ya que esta ha demostrado una alta eficiencia en la separación de datos con alta variabilidad y con distribuciones no lineales; un ejemplo de ellos se presenta en la Figura 6.4; para este ejemplo, se presenta un conjunto de datos generados de manera artificial ejemplificando un problema complejo de separación no lineal. El método de t-SNE logra agrupar los datos en las diferentes clases requeridas, maximizando además la separabilidad entre las clases.

El método de t-SNE tiene un desempeño sobresaliente al momento de procesar los datos tanto lineales como no-lineales, sin embargo, tiene el inconveniente de ser un método de aplicación única, es decir, si nuevos datos son agregados al sistema, se debe de ejecutar nuevamente. En este trabajo doctoral, se ha propuesto una modificación al método t-SNE, agregando una arquitectura neuronal con neuronas spiking de respuesta sigmoidal. Al agregar esta arquitectura, las neuronas se encargan de sintetizar las características esenciales de los datos y, si nuevos datos son anexados, las neuronas tienen la capacidad de adaptarse a los posibles cambios generados por los nuevos datos, modificando sus valores de conductancias sinápticas. Por lo tanto, agregar esta arquitectura neuronal mejora el desempeño de sistema t-SNE para procesamiento de datos.

6.7. Conclusiones del Capítulo

En este capítulo, se ha presentado una arquitectura neuronal denominada autoenconder, que logra la reducción dimensional de los datos. El autoencoder tiene la limitante de poder procesar únicamente datos con distribuciones lineales.

Se ha presentado el método de reducción dimensional denominado t-SNE. Este método es robusto en la extracción y reducción de características de datos tanto lineales como no lineales, por lo que su uso en el campo de la reducción dimensional se ha extendido.

Se ha propuesto una modificación al método de t-SNE, agregando una arquitectura neuronal con neuronas de tipo spiking y respuesta de tipo sigmoide; esta modificación hace más robusto al método t-SNE, dotándolo de una adaptabilidad ante la presencia de nuevos datos.

7 Resultados

En este capítulo se presentan los resultados obtenidos de la tarea de reducción dimensional desarrollado por la red neurona spiking de tipo no lineal propuesta en este trabajo. Las pruebas de desempeño se realizaron sobre las bases de datos referentes a "Escritura de Números", el conjunto de imágenes de "Frutas, Flores y Rostros", y "Tierras de Cultivo". Estas bases de datos son descritas a detalle en el capítulo 5. Finalmente, se incluye la sección de análisis de resultados donde se discute la pertinencia de los resultados obtenidos.

7.1. Eficiencia de Calidad Relativa en la Reducción Dimensional

Con la finalidad de determinar la eficiencia de calidad relativa en la Reducción Dimensional lograda con la Red Neuronal Spiking (RNS) y determinar lo compacto de las agrupaciones generadas, se calcula el Error Cuadrático Medio (ECM) de las distancias de cada muestra que forma una agrupación y el propio centro de la clase, con la finalidad de determinar la dispersión de cada una de las agrupaciones generadas.

En principio, una RNS puede generar M clases con dimensiones reducidas, de un conjunto de vectores en una base de datos que también contenga M clases, además de poder visualizar en un gráfico la distribución final de los vectores. Para nuestro caso, se generan gráficos en 3D con la finalidad de apreciar las agrupaciones resultantes; los espacios dimensionales de salida son etiquetados como Dimensión 1, 2 y 3. La calidad del desempeño de la reducción de dimensiones lograda con la RNS sobre M clases es evaluada con el operador de ECM, de acuerdo con la Ecuación (7.1).

$$ECM = \frac{1}{N} * \sum_{i=1}^{N} \left(D_{1,i} - D_{2,i}\right)^2, \; with \; i = 1, ..., N \tag{7.1}$$

Los parámetros involucrados en la Ecuación (7.1) son presentados en la Tabla 7.1 y se han considerado los cálculos de comparación de similitudes entre una distribución de referencia y las producidas por una distribución aleatoria y la RNS. Primero, se realiza una "Distribución de Referencia", con una neurona clásica donde se conocen sus parámetros de manera previa. Como segundo paso, se genera una distribución de los datos de manera aleatoria con una función de distribución uniforme, y finalmente se obtiene la distribución de los datos aplicando la RNS. Con el objetivo de determinar la eficiencia relativa en la Reducción Dimensional lograda con la RNS, se calcula la dispersión de las agrupaciones generadas y se compara con la distribución de referencia. Un procedimiento similar se realiza entre la distribución aleatoria y nuevamente la distribución de referencia; de esta manera, se demuestra que las distribuciones que se generan con las RNS tienden a ser similares a las distribuciones de referencia y que su desempeño es constante.

Tabla 7.1: Parámetros del Experimento

Parámetros
N, número de muestras en una clase.
$\{D_{1,i}\}$, conjunto de distancias entre el centro de la clase y todas las muestras que la conforman. Agrupación aleatoria con distribución uniforme
$\{D_{2,i}\}$, conjunto de distancias entre el centro de la clase y todas las muestras que la conforman. Agrupación con la RNS

Con lo descrito anteriormente, se desarrollan dos cuantificaciones de ECM.

- ECM-1 RNS vs. Referencia. Medida de desviación de la distribución obtenida de RNS versus la distribución de referencia.
- ECM-2 Aleatoria vs. Referencia. Medida de desviación de la distribución obtenida de forma aleatoria versus la distribución de referencia.

Comparando ECM-1 con ECM-2, es de esperar que se obtengan valores mas bajos para ECM-1 que con ECM-2, lo cual comprueba la eficiencia y constancia de las agrupaciones generadas por la RNS. Los resultados de estos valores, para cada una de las bases de datos analizadas, son presentadas en gráficos de barra para una mejor comprensión.

7.2. Matriz de co-Ranking

La matriz de *co-Ranking* es empleada como un segundo método de medición de la calidad de la reducción de dimensionalidad, comparando la distribución de los datos originales contra la

distribución generada.

Dada la reducción de dimensionalidad, queda por completar la evaluación del nuevo mapeo de datos sobre los datos originales expresados en dimensiones altas. Para determinar esta medición de manera objetiva, se propone utilizar la matriz de *co-Ranking*, tomada de [79], aplicando la Ecuación (7.2).

$$Q_{kl} = |\{(i,j) \mid \rho_{ij} = k \;\; y \;\; r_{ij} = l\}| \tag{7.2}$$

En donde ρ_{ij} representa el rango de distancia de cada característica de la muestra x_i con respecto a la muestra x_j en la base de datos con el espacio de alta dimensión. Por su parte, $r_{i,j}$ es el rango de distancias de la característica de la muestra y_i con respecto a la muestra y_j en el espacio propuesto de baja dimensión y $|\cdot|$ denota el número total de muestras del conjunto de datos. Cada muestra x_i tiene su correspondiente muestra en y_i, por lo tanto, el rango o distancias de las muestras originales deben ser, para un caso ideal, idénticas a los rangos de las muestras propuestas en la salida. Los rangos totales son calculados empleando la Ecuación (7.3), en donde los valores de $\delta_{i,k}$ representan el valor de distancia de x_i a x_j, mientras que d_{ij} lo representa para y_i a y_j.

$$\begin{aligned} \rho_{ij} &= |\{k \mid \delta_{ik} < \delta_{ij} \; or \; (\delta_{ik} = \delta_{ij} \; and \; k < j)\}| \\ r_{ij} &= |\{k \mid d_{ik} < d_{ij} \; or \; (d_{ik} = d_{ij} \; and \; k < j)\}| \end{aligned} \tag{7.3}$$

La Ecuación (7.2), genera la matriz de co-Ranking; cuando la diagonal principal contiene los valores mayores del cálculo, esto se interpreta como un mapeo o correspondencia perfecta entre las distribuciones de entrada y de salida. Aunque esta situación es casi imposible de cumplir, ya que las muestras originales al ser procesadas con las expresiones matemáticas de la reducción dimensional sufren alteraciones, estas alteraciones son interpretadas como intrusiones o extrusiones. Se considera una intrusión cuando el punto j tiene un rango más bajo con respecto al punto i, esto para la representación de baja dimensión en comparación con los datos de alta dimensión. Por el contrario, cuando una extrusión ocurre, un punto j tiene un rango más alto con respecto al punto i en la representación de baja dimensión en comparación con los datos de alta dimensión. Los puntos de intrusión están ubicados debajo de la diagonal principal; los puntos de extrusión por arriba.

En [80], se presenta una mejora para el cálculo de la matriz de co-Ranking, al considerar que

la calidad de la representación de datos en el espacio de baja dimensión, se mide en función del número de puntos que permanecen dentro de una vecindad k durante el proceso de reducción. Para realizar esta modificación se emplea la Ecuación (7.4), donde N representa el número de puntos totales y K determina el valor de la vecindad.

$$Q_{NX}(K) = \frac{1}{KN} \sum_{k=1}^{K} \sum_{l=1}^{K} Q_{kl} \tag{7.4}$$

$Q_{NX}(K)$ es sensible para agrupaciones con una pequeña cantidad de muestras, que son los casos donde el error de mapeo suele aumentar. La relación de $Q_{NX}(K)$ versus la cantidad de muestras sigue manteniendo, para un caso ideal, una curva asentada en la diagonal principal. Para cada base de datos tratada, se presentan las gráficas de la distribución de referencia y la distribución obtenida de la RNS.

7.3. Análisis de Resultados

En esta sección se presentan los resultados obtenidos al procesar las tres bases de datos que se han propuesto de manera inicial. Para cada base de datos se presenta, primeramente, las imágenes con las reducciones para 3 dimensiones y su comparación con la distribución de referencia. Se muestran además, las gráficas de los centros de las clases para ambos modelos: referencia y RNS. Finalmente, se presentan las gráficas del cálculo de la matriz de co-Ranking. Se ha buscado que en conjunto las bases de datos sean distintas en cuanto a clases y cantidades en las características a minimizar, con la finalidad de demostrar la pertinencia de la propuesta desarrollada en este trabajo de investigación.

7.3.1. Procesamiento de Base de Datos de Números Escritos a Mano.

La primera base de datos procesada por el sistema descrito en la sección anterior, es la base de datos que corresponde a la escritura de forma manual de números del 0 al 9. El resultado de separación general y la de cada clase es presentada en la Figura 7.1. Como punto importante de las distribuciones en esta gráfica, es que se logra reducir las dimensiones en los datos de salida, manteniendo la separabilidad en cada clase. Para esta base de datos se tienen 10 clases que corresponden a cada número.

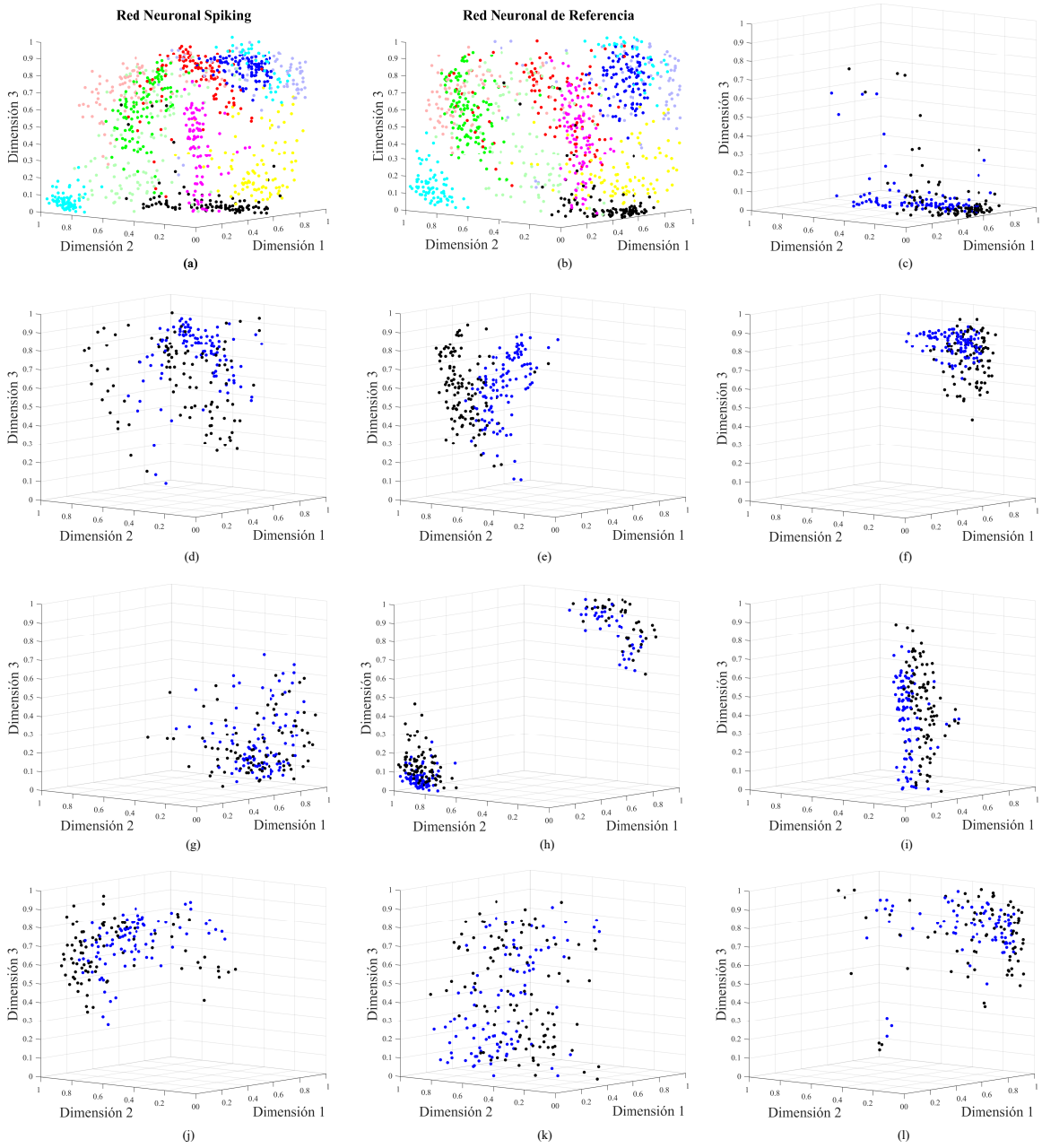

Figura 7.1: Resultado del Proceso de Reducción Dimensional con RNS aplicado a la Base de Datos de Números Escritos a Mano. (a) Distribución con RNS. (b) Distribución de Referencia. (c) Clase número 0. (d) Clase número 1. (e) Clase número 2. (f) Clase número 3. (g) Clase número 4. (h) Clase número 5. (i) Clase número 6. (j) Clase número 7. (k) Clase número 8. (l) Clase número 9.

En la Figura 7.2, se muestran las gráficas de los valores de los centros de cada clase. En la Figura 7.2(a) se muestran los valores del centro de cada una de las 10 clases para la Dimensión 1, correspondiente a los números del 0 – 9. Se puede notar que la distribución que se logra con la RNS es similar a la distribución de referencia.

Como se indicó en la sección 7.1, y para evaluar la calidad de la tarea de la reducción de

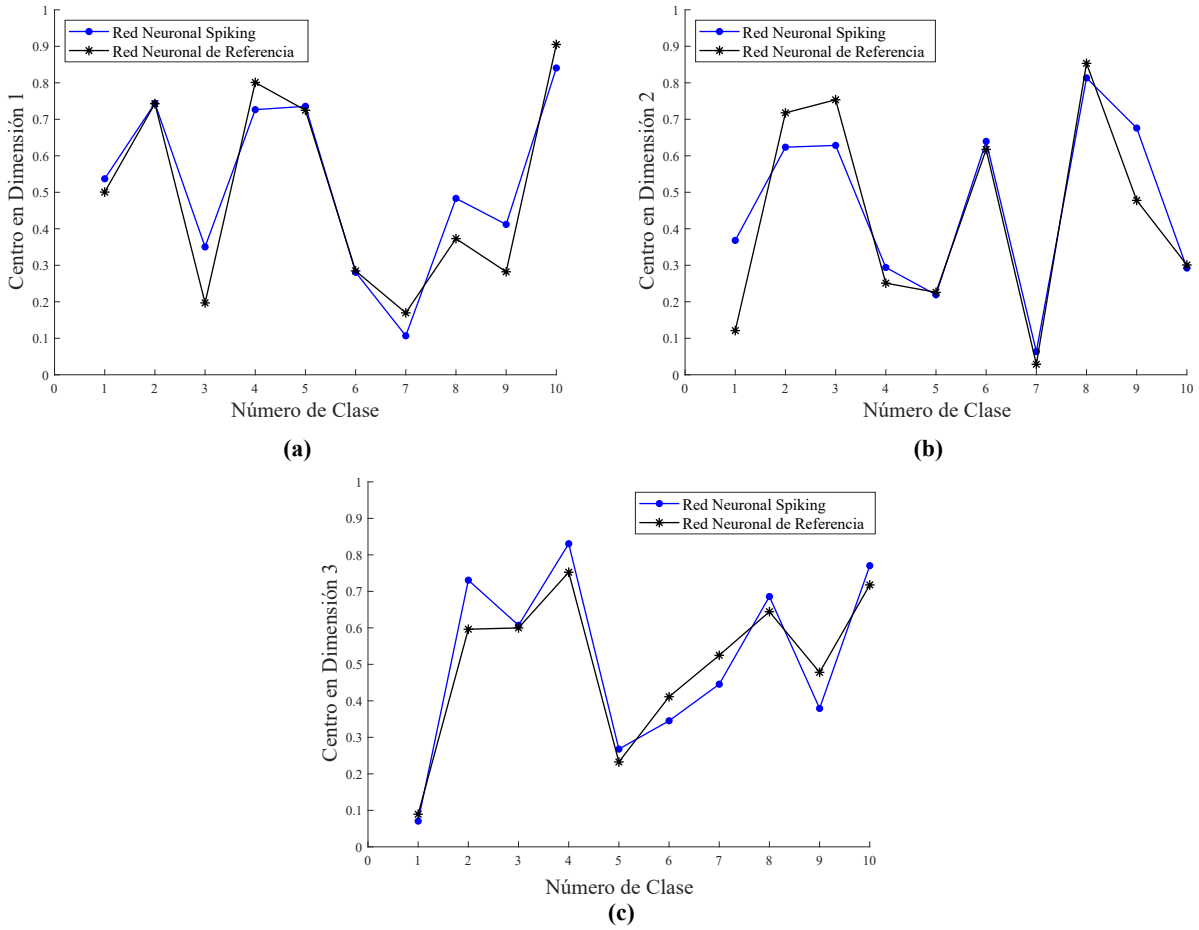

Figura 7.2: Valores de los centros para las 10 clases que corresponden a los números de 0 a 9. (a) Valores de los centros en la Dimensión 1. (b) Valores de los centros en la Dimensión 2. (c) Valores de los centros en la Dimensión 3.

dimensionalidad, para la base de datos de números escritos a mano, se calculan los valores de ECM-1 y ECM-2. Los gráficos en el formato de barras mostrados en la Figura 7.3, corresponden a la comparación de la distribución de referencia con la RNS y la distribución aleatoria con función de distribución normal, respectivamente. De estos gráficos, se resalta que los valores para ECM-1 son pequeños en comparación a los valores de ECM-2, indicando que la distribución de RNS es en gran medida semejante a las distribuciones de referencia.

Finalmente, para la base de datos de números escritos a mano, se presentan las gráficas obtenidas del cálculo de la matriz de co-Ranking. Con estas gráficas se puede evaluar la calidad de la representación en baja dimensión. Las gráficas de la matriz de co-Ranking son presentadas en la Figura 7.4. En estas gráficas se observa el desempeño de la distribución de referencia, la distribución por RNS y la obtenida por el método de t-SNE. Recordemos que un gráfico sobre la diagonal principal representaría un caso idealizado. La gráfica que corresponde la distribución RNS (gráfico en color azul), es continua y sin variaciones en su trayectoria y se mantiene apegada

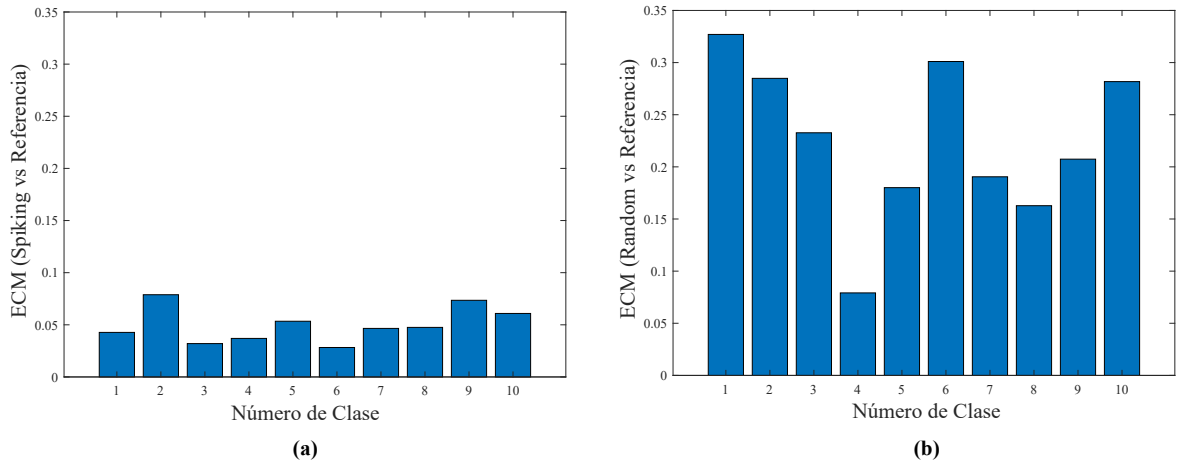

Figura 7.3: Valores del Error Cuadrático Medio. (a)ECM-1 RNS versus Referencia. (b)ECM-2 Distribución Aleatoria versus Referencia.

a la gráfica de la distribución de referencia. Para el caso del t-SNE, el gráfico presenta variaciones en el rango de 0 a 300 del valor de K, indicando que el proceso de mapeo para las muestras iniciales no es adecuado.

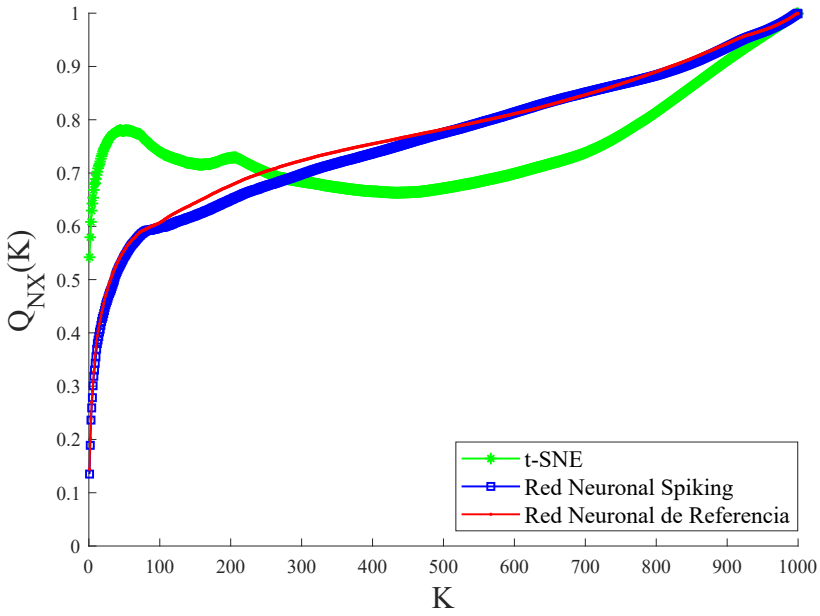

Figura 7.4: Gráficas de la matriz de co-Ranking

7.3.2. Procesamiento de Base de Datos de Frutas, Flores, y Rostros.

La segunda base de datos procesada corresponde a imágenes de Frutas, Flores, y Rostros. Las imágenes se encuentran en tonalidades de gris con una dimensión de 100 filas por 100 columnas y 256 niveles de intensidad para cada pixel. Estas imágenes han sido procesadas y de ellas se han extraído componentes basadas en la textura y estructura de los objetos contenidos. Se parte con

la premisa de que los objetos de interés tendrán valores diferentes en textura y estructura. Para la extracción de características se empleó la técnica de procesamiento de imágenes nombrada *Patrones Binarios Locales (LBP)*. La base de datos queda conformada de la siguiente manera: Contiene 3 clases de objetos y 67 características en cada vector de datos; a estos vectores se les aplicó el proceso de reducción dimensional y se reducen a 3 componentes. El resultado del procesamiento se observa en la Figura 7.5.

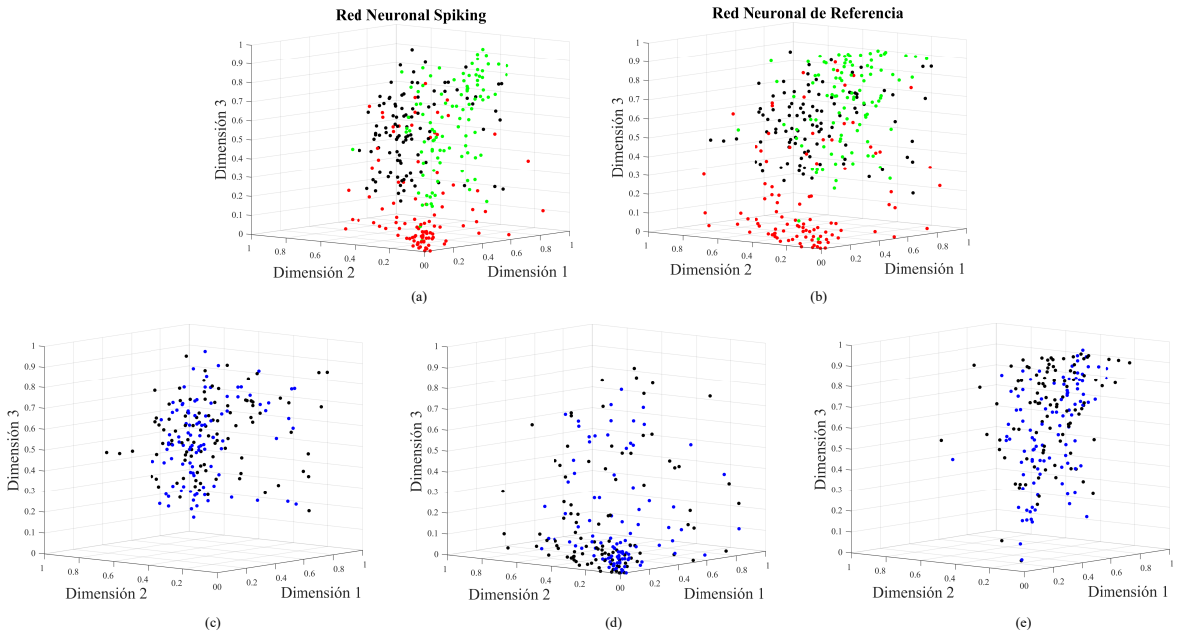

Figura 7.5: Resultado del Proceso de Reducción Dimensional con RNS aplicado a la Base de Datos de Frutas, Flores y Rostros. (a) Distribución con RNS. (b) Distribución de Referencia. (c) Clase Frutas. (d) Clase Flores. (e) Clase Rostros.

Los valores numéricos de los centros de las clases de la distribución de referencia son presentados en la Figura 7.6, y se comparan con los centros obtenidos por la RNS. Es de notar que para las Dimensiones 1 los valores para las clases 2 y 3 llegan a ser iguales, lo que indica que la distribución generada por la arquitectura neuronal propuesta es idéntica a la distribución de referencia.

El gráfico de barras que se muestra en la Figura 7.7 muestra de manera numérica la semejanza entre la distribución de referencia y la distribución de la RNS; para la clase correspondiente a Frutas, clase número 1 las distribuciones son casi idénticas, con un ECM del orden de 0.05. Para la clase de flores y rostros se conserva un valor de semejanza ECM por debajo de 0.1.

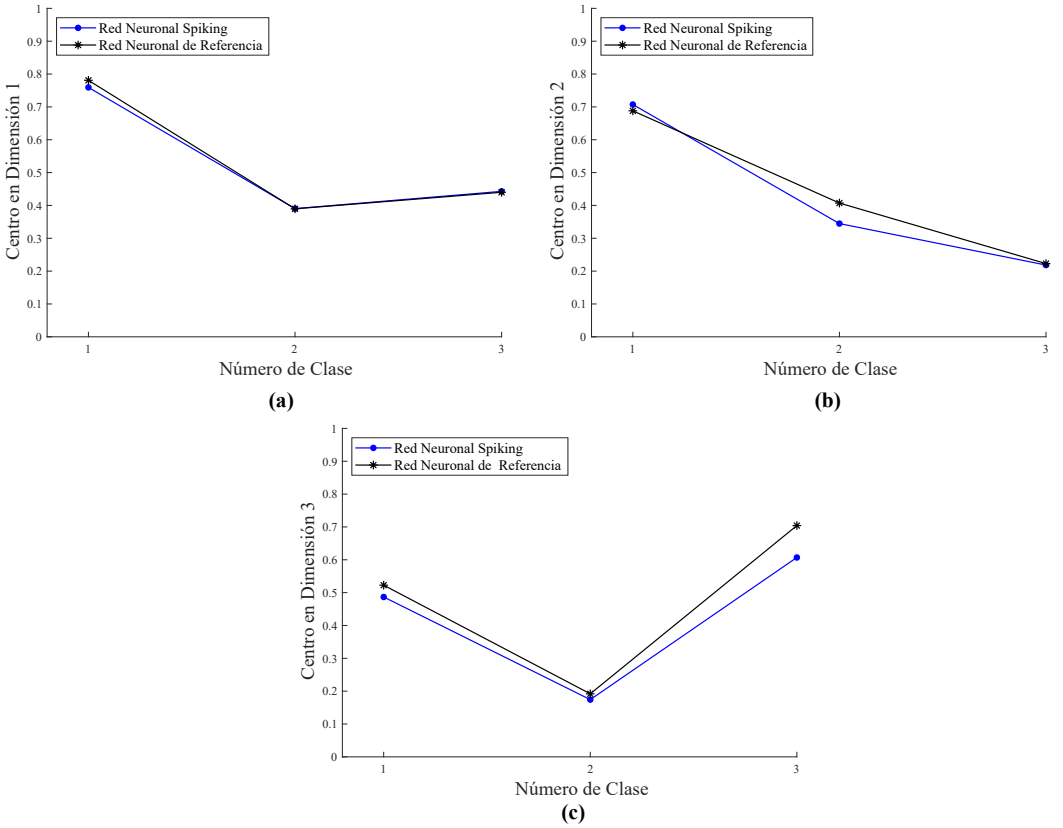

Figura 7.6: Valores de los centros para las 3 clases que corresponden a Frutas, Flores y Rostros. (a) Valores de los centros en la Dimensión 1. (b) Valores de los centros en la Dimensión 2. (c) Valores de los centros en la Dimensión 3.

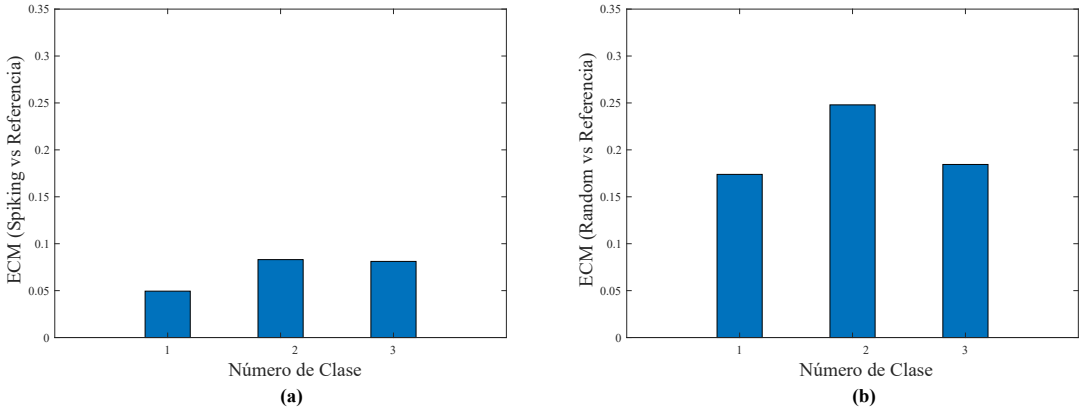

Figura 7.7: Valores del Error Cuadrático Medio. (a)ECM-1 RNS versus Referencia. (b)ECM-2 Distribución Aleatoria versus Referencia.

Con la finalidad de determinar la calidad de la representación de los datos en baja dimensión se ha calculado la matriz de co-Ranking para la distribución de referencia, la distribución obtenida con la RNS y el método de t-SNE. Los resultados son presentados en la Figura 7.8. De los tres gráficos, el gráfico de color azul, que corresponde al resultado con la RNS, nuevamente se mantiene continuo y es el que más se aproxima a la diagonal principal.

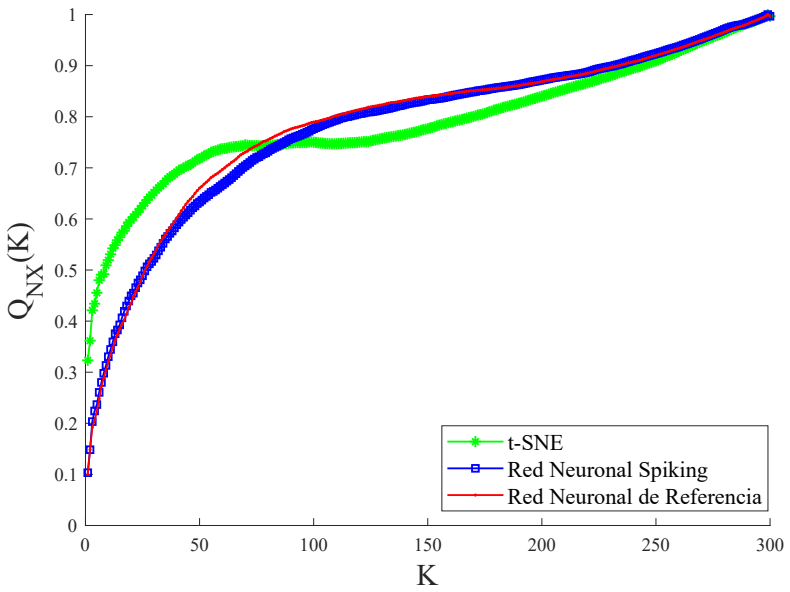

Figura 7.8: Gráficas de la Matriz de co-Ranking para la Base de Datos de Frutas, Flores y Rostros

7.3.3. Procesamiento de Base de Datos de Tierras de Cultivo.

Para la tercera prueba, se ha procesado la base de datos que corresponde a tierras de cultivo. En ella se tienen 7 clases que corresponden a los cultivos de: 1 -Maíz; 2 -Guisantes; 3 -Canola; 4 -Soya; 5 -Avena; 6 -Trigo; y 7 -Cultivos de hoja ancha. Para cada muestra, de las 7 clases, se tienen 174 características que son reducidas a tan sólo 3, por la técnica propuesta. Después de realizar el procesamiento para la base de datos, se obtuvieron las gráficas que se presentan en la Figura 7.9.

La Figura 7.10 contiene los gráficos donde se comparan los valores numéricos de los centros de las 7 clases entre la distribución de referencia y la RNS; es importante notar que para la Dimensión 2, las clases 4, 5, y 6 tienen valores coincidentes, lo cual indica la alta similitud entre la distribución de datos de referencia y la obtenida con la arquitectura neuronal propuesta. Para la Dimensión 3 se presenta esta situación en las clases 3, 4 y 7. Para el resto de las clases la diferencia es muy pequeña, lo cual indica la alta similitud entre las dos distribuciones.

Se obtuvieron los valores de ECM-1, que compara la distribución de referencia contra la distribución de la RNS y el mismo procedimiento sobre ECM-2 considerando la distribución aleatoria. Las gráfica en formato de barras es presentado en la Figura 7.11

Las gráficas presentadas en la Figura 7.12 son obtenidas del cálculo de la matriz de co-Ranking para la base de datos de Tierras de Cultivo. Nuevamente, se observa que los datos de salida tienen buen desempeño por lo que se considera que la reducción Dimensional mantiene la

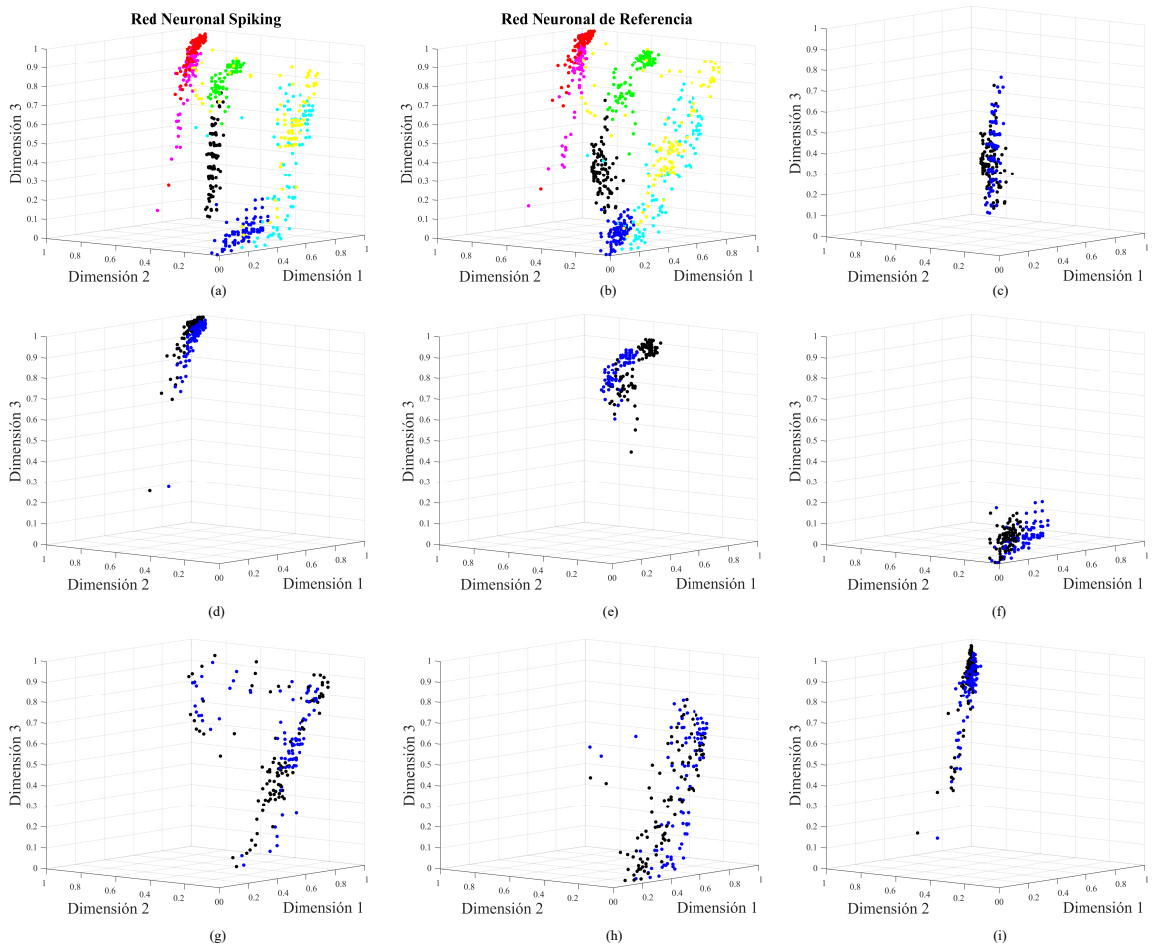

Figura 7.9: Resultado del Proceso de Reducción Dimensional con RNS aplicado a la Base de Datos de Tierras de Cultivos. (a) Distribución con RNS. (b) Distribución de Referencia. (c) Clase Cultivo de Maíz. (d) Clase Cultivo de Guisantes. (e) Clase Cultivo de Canola. (f) Clase Cultivo de Soya. (g) Clase Cultivo de Avena. (h) Clase Cultivo de Trigo. (i) Clase Cultivo de Hoja Ancha.

distribución de los datos originales.

Los resultados presentados en esta sección, para las tres bases de datos procesadas, avalan la buena respuesta que tiene el sistema al momento de realizar el proceso de reducción de dimensionalidad, manteniendo la similitud de las distribuciones en los datos de salida respecto a los datos originales. Además, la técnica t-SNE que originalmente fue planteada para la visualización de los datos, ha servido como plataforma para el procesado de los datos y demostrar un nuevo uso de las neuronas biológicamente plausibles.

7.4. Análisis de los Resultados

Los resultados presentados en esta sección, para las tres bases de datos procesadas, avalan la buena respuesta que tiene el sistema al momento de realizar el proceso de reducción de

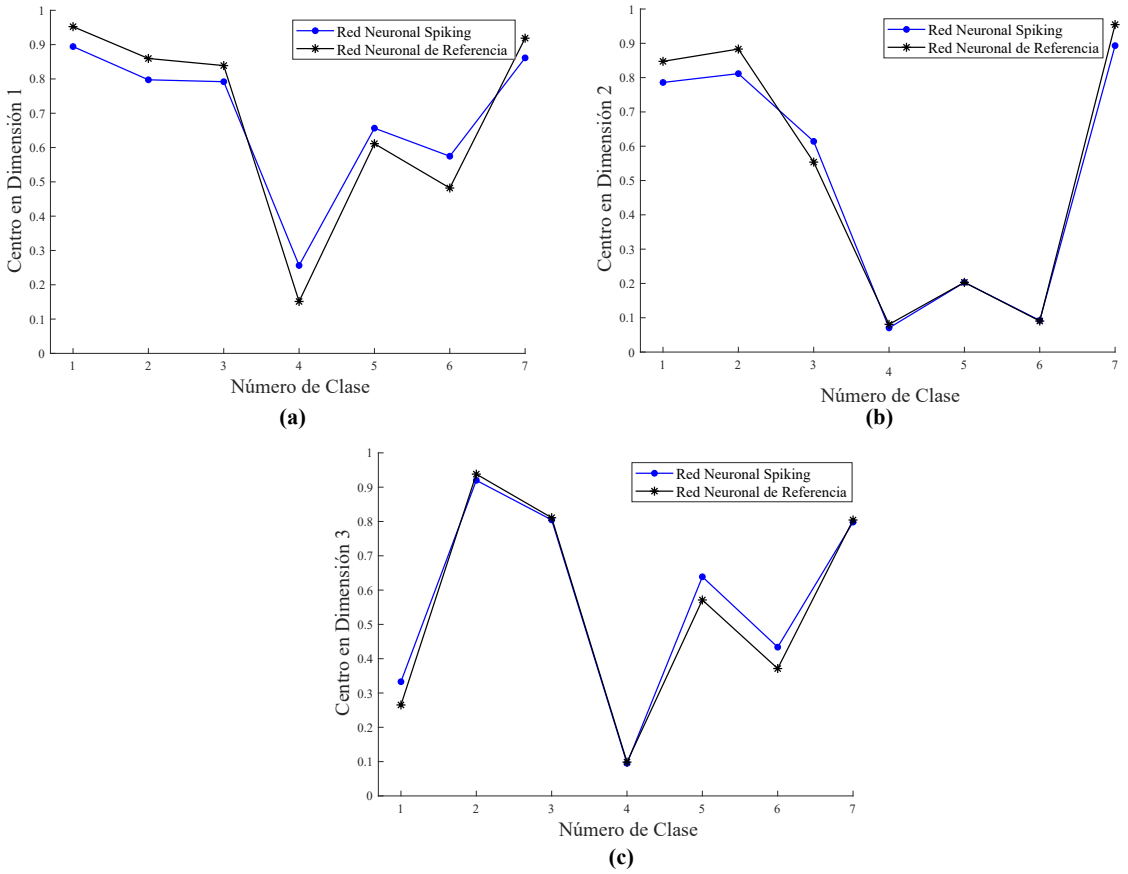

Figura 7.10: Valores de los Centros de las Clases que Corresponden a 7 Tipos de Cultivos. (a) Valores de los centros en la Dimensión 1. (b) Valores de los centros en la Dimensión 2. (c) Valores de los centros en la Dimensión 3.

dimensionalidad, manteniendo la similitud entre las distribuciones de los datos de salida respecto a los datos originales.

Para la primera base de datos, referente a los números escritos a mano, podemos mencionar que se ha elegido debido al reto que presenta, por las similitudes que existen entre las personas al generar los trazos en la escritura de cada número. Con los resultados obtenidos se puede argumentar que en todas las clases se ha logrado la reducción en la dimensión de sus características, manteniendo la separación entre dichas clases.

La clase que presentó el mayor valor en la "medición de calidad relativa", fue la clase que corresponde al número dos, con un valor de 0.084, lo que representó el caso de mayor diferencia, sin embargo, este valor es bajo indicando que existe una alta semejanza entre la distribución de los datos que corresponden a la red neuronal de referencia contra la red neuronal spiking. El resto de los valores para las otras clases, se pueden observar de las gráficas de la Figura 7.3; estos valores son aún más bajos indicando claramente que de manera global, la red neuronal spiking es eficiente en la tarea de la reducción dimensional.

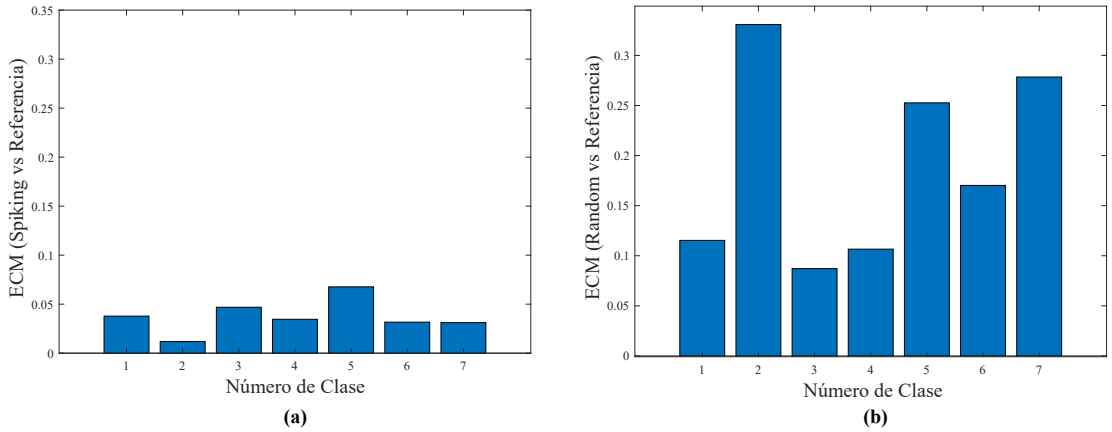

Figura 7.11: Valores del Error Cuadrático Medio. (a)ECM-1 RNS versus Referencia. (b)ECM-2 Distribución Aleatoria versus Referencia.

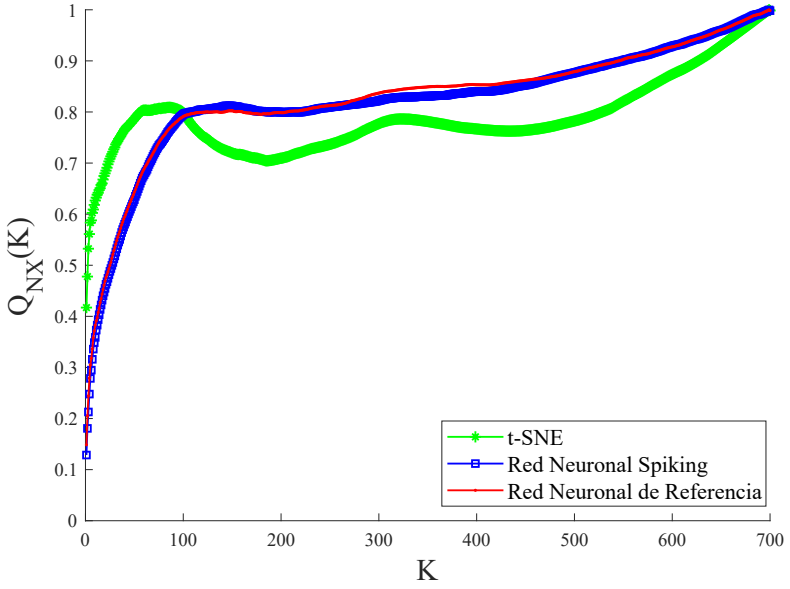

Figura 7.12: Gráficas de la Matriz de co-Ranking para la Base de Datos de Tierras de Cultivo

Las gráficas presentadas en la Figura 7.4 corresponden al cálculo de la matriz de co-Ranking e indican la calidad en la reducción para esta base de datos. Es de notar la semejanza entre la gráfica de los datos de referencia y la obtenida por la arquitectura neuronal spiking indicando, nuevamente, lo eficiente de la arquitectura propuesta en el resultado de la reducción de dimensión de los datos.

Las bases de datos que corresponden a frutas, flores y rostros, son consideradas en este trabajo por dos razones: la primera de ellas, es que cada muestra es una imagen, por lo que tiene una distribución estructural y cada valor de un pixel o componente de una muestra tiene una relación establecida con los pixeles aledaños. La segunda es que los objetos que se encuentran dentro de las imágenes, comparten niveles de colores y geometrías semejantes, por lo que es de

esperar una alta mezcla en la representación de los datos, representado así un base de datos compleja con una clasificación de tipo no lineal.

Las gráficas de la Figura 7.7, demuestran una notoria semejanza entre las distribuciones de referencia y la distribución lograda con la arquitectura neuronal spiking. Los valores en la "medición de calidad relativa" para las tres clases son muy bajos indicando una semejanza considerable.

Las gráficas de la matriz de co-Rankin, en la Figura 7.8, muestran que el método t-SNE tiene un desempeño ligeramente menor en comparación con las distribuciones de referencia y la lograda por la arquitectura neuronal spiking. Es importante recordar que el método t-SNE debe ser ejecutado cada ocasión que un nuevo dato es anexado a la base de datos.

Finalmente, se consideró la base de datos de tierras de cultivo, de las cuales se han extraído 174 características, realizando mediciones de longitudes de onda sobre regiones con diferentes cultivos. Esta base de datos presenta el reto de lograr la compresión a tan solo 3 características que mantengan la distribución de los datos originales.

Las gráficas de la "medición de calidad relativa", de la Figura 7.11, demuestran que es posible la reducción dimensional manteniendo una alta semejanza con los valores de referencia.

Las gráficas que corresponden al cálculo de la matriz de co-Ranking de la base de datos correspondientes a tierras de cultivo, dan la veracidad de que la reducción de dimensionalidad es correcta considerando la alta tasa de reducción dimensional al extraer la información pertinente de 174 características y representarlas en tan solo 3.

7.5. Conclusiones del Capítulo

Se han elegido tres bases de datos bajo la consideración de la gran cantidad y variabilidad en los datos. Cada una de ellas presenta retos diferentes sobre la cantidad de características y el ambiente de donde provienen. En todas ellas, el sistema desarrollado ha demostrado un alto desempeño en la extracción de características esenciales y la reducción de dimensión.

Se ha planteado el cálculo de la "medición de calidad relativa" para determinar la semejanza entre la distribución de datos de referencia y la obtenida por la arquitectura neuronal spiking desarrollada en este trabajo doctoral. Los resultados obtenidos demuestran la alta semejanza entre las dos distribuciones.

El cálculo de la matriz de co-Raking se ha empleado para determinar la calidad de la reduc-

ción dimensional para tres bases de datos y los resultados avalan que la reducción lograda por el sistema desarrollado en este trabajo doctoral es de alta eficiencia.

Además, la técnica t-SNE que originalmente fue planteada para la visualización de los datos, ha servido como plataforma para el procesado de los datos y demostrar un nuevo uso de las neuronas biológicamente plausibles.

8 Conclusiones y Perspectivas del Trabajo Desarrollado

8.1. Conclusiones

En este trabajo se han presentado propuestas para el análisis de datos, el uso de redes neuronales de tercera generación o neuronas spiking y la proyección de paradigmas emergentes como el internet de las cosas; durante el desarrollo se han extraído conclusiones que son presentadas a continuación:

- El internet de las cosas (IDC) se encuentra en plena consolidación y día con día se suman ambientes de aplicación. Este paradigma ha demostrado capacidad, haciendo que los procesos mejoren en tiempo, se reduzcan los recursos computacionales y se tomen decisiones adecuadas, logrando de esta manera la reducción en el costo de operación de sistemas, que incluyen la dinámica de seres humanos y eventos complejos adjunto. El principal problema en este paradigma es el análisis de grandes cantidades de información y ya que las técnicas clásicas utilizadas en su diseño fueron rebasadas, se hace necesario la implementación de nuevas técnicas que tengan esta directriz. En el trabajo desarrollado, se han considerado y analizado tres bases de datos provenientes de sistemas IDC, tomando en cuenta la cantidad de información y su procesamiento, y se ha logrado su análisis para la extracción de información y reducción de dimensión.
- El término de Big Data, más allá de representar grandes cantidades de información, es aplicado al almacenamiento estructurado de datos, lo que permite utilizarlo con el objetivo de buscar patrones de conducta o lineamientos que puedan establecerse para poder generar predicciones. Una de las premisas para el Big Data, es que no debe limitarse a una sola fuente como suministro de información, por lo que en este trabajo se ha considerado que los datos provengan de diferentes ambientes, logrando con esto demostrar la robus-

tez del trabajo implementado y su versatilidad, siguiendo un nuevo esquema dedicado al procesamiento de grandes cantidades de información.

- Los datos dan un panorama inicial del comportamiento de un ambiente o espacio de trabajo, según los eventos que se registren en él; sin embargo, existen eventos que dependen de la generación de otros eventos generados en otros espacios de trabajo, estos eventos son complejos y hacen que la tarea de análisis de datos se convierta en algo abstracto para enfrentar. Una estrategia comúnmente aceptada y que es distintiva de forma particular en los sistemas inteligentes, es encontrar características relevantes que definan a los datos y desechar las características que aporten poca información. Lo anterior en este trabajo permitió reducir la cantidad de información a procesar y el tiempo para presentar un resultado.
- Uno de los problemas relevantes en el análisis de sistemas complejos es la búsqueda de parámetros que logren optimizar a estos sistemas, esta tarea se ha abordado desde diferentes perspectivas. Actualmente, se realizan estudios sobre el comportamiento organizacional de seres vivos en la naturaleza y de este análisis, se han generado algoritmos computacionales con la capacidad de resolver en forma competitiva y eficiente problemas de optimización. En este trabajo, se ha presentado el análisis sobre tres algoritmos metaheurísticos bioinspirados: (1) Algoritmo de Optimización por Hormigas, (2) Algoritmo de Organización por Enjambre de Partículas y (3) Algoritmo de Colonia Artificial de Abejas. Los tres algoritmos se han implementado y evaluado con tres actividades diferentes: resolver una función matemática, el análisis y agrupamiento de datos y el procesamiento de una imagen. Con base en los resultados, se ha determinado que el método Colonia Artificial de Abejas presentó el mejor desempeño y por lo tanto, este método se consideró para desarrollar el proceso de optimización del cálculo de conductancias para las neuronas spiking.
- El algoritmo de Colonia Artificial de Abejas, se implementó para buscar los parámetros de una neurona spiking logrando que ésta tenga una respuesta de manera sigmoidal, lo cual representa un nuevo comportamiento no reportado a la fecha en la literatura.
- El estudio de las neuronas spiking ha tomado gran interés para aplicaciones en la rama de la inteligencia artificial, ya que se ha demostrado que son más apegadas al comportamiento de las neuronas biológicas del cerebro. En este trabajo se ha implementado una arquitectura con neuronas spiking de respuesta no lineal y se ha demostrado que la neurona no lineal mejora la capacidad computacional de este tipo de arquitecturas para abordar problemas

cuya solución es no linealmente alcanzable con la arquitectura original.

- La reducción dimensional es una tarea muy empleada para el tratamiento y extracción de información en bases de datos. Un modelo ampliamente utilizado para la visualización de los datos en baja dimensión es el t-SNE. En este trabajo se ha propuesto utilizar t-SNE, que implica insertarlo en el modelo una arquitectura neuronal que emplea neuronas spiking de tipo sigmoidal y se ha demostrado que tiene la capacidad de generar distribuciones en dimensiones reducidas conservando las distribuciones de los datos iniciales.
- Se ha comparado la respuesta del sistema desarrollado contra un sistema de referencia. Para una evaluación cuantitativa se han considerado dos métricas: el cálculo de la matriz de co-Ranking y el cálculo de la medición de calidad relativa. Los resultados obtenidos demuestran lo eficiente que es el sistema de reducción dimensional desarrollado en este trabajo doctoral.
- Con los experimentos desarrollados en este trabajo de doctorado, se demuestra que las aportaciones son originales y que éstas contribuyen al conocimiento en las áreas de: (1) Procesamiento y extracción de información de bases de datos que complementan el desarrollo de la tecnología del internet de las cosas. (2) Metaheurística para la optimización de problemas complejos. (3) Aplicación de sistemas neuronales de tercera generación o spiking.
- Finalmente, concluimos que el trabajo de doctorado, cuyo desarrollo temático se presenta en el manuscrito de esta tesis, ha alcanzado todos los Objetivos: El General y Los Particulares.

8.2. Perspectivas

Las perspectivas o proyecciones de este trabajo de doctorado, son las siguientes:

- Si bien el método de t-SNE es eficiente en la visualización de los datos en baja dimensión, otros paradigmas alternativos de procesamiento informático podrán ser abordados.
- La nueva neurona spiking con respuesta sigmoidal es originalmente propuesta en este trabajo, por lo que es atractivo determinar su desempeño aplicándola a problemas en diferentes ámbitos.
- Dada la información disponible en sistemas para el procesamiento de imágenes digitales en la tarea de mejora de contraste de los niveles de color, es posible implementar arquitecturas

de redes neuronales spiking con respuesta sigmoidal, que tengan la capacidad de desarrollar esta tarea.

- Con la experiencia adquirida por el Grupo de Sistemas VLSI de la Sección de Electrónica del Estado Sólido, la cual incluye la implementación de neuronas artificiales spiking en hardware, puede afirmarse entonces que el estudio y desarrollo de la neurona spiking sigmoidal está latente.

8.3. Productos Derivados del Trabajo

Los productos derivados durante el desarrollo de este trabajo doctoral son:

- *Spiking Neural Network Architecture Comparison by Solving the Non-linear XOR Problem.* Artículo publicado en extenso en las memorias del 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2020.

- *Approaching Optimal Nonlinear Dimensionality Reduction by a Spiking Neural Network.* Artículo publicado en la revista: Electronics MDPI 2021, 10(14), 1679, con número de registro doi.org/10.3390/electronics10141679. Received: 21 May 2021 / Revised: 7 July 2021 / Accepted: 7 July 2021 / Published: 14 July 2021. Impact Factor : 2.397. Cite Score: 2.7 Scopus.

Apéndices

8.4. Apéndice A: Métodos Metaheurísticos

8.4.1. Código para la Técnica ACO-R

8.4.2. Código para la Técnica PSO

8.4.3. Código para la Técnica ABC

8.5. Apéndice B: Neurona Spiking Sigmoidal

8.5.1. Código para la Búsqueda de Parámetros de la Neurona Spiking No Lineal

Los códigos para cada uno de los apéndices son recopilados en el siguiente enlace:

https://drive.google.com/drive/folders/1uuYwoShRw5sjvpR8XKwBEVPGTSrVB6tK?usp=sharing

En caso de fallas en la dirección del enlace, el autor sugiere el envío de un correo electrónico solicitando los apéndices a la cuenta de correo: alvaro.anzueto.rios@gmail.com

8.6. Apéndice C: Evaluación Anti-Plagio

8.6.1. Resultado de similitud calculado por sistema Turnitin: 5 %

Bibliografía

[1] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] V. Govindasamy and P. Thambidurai, "Probabilistic fuzzy logic based stock price prediction," *International Journal of Computer Applications*, vol. 71, no. 5, 2013.

[3] F. Sari and Y. Latief, "Safety cost estimation of building construction with fuzzy logic and artificial neural network," in *Journal of Physics: Conference Series*, vol. 1803, pp. 01–10, IOP Publishing, 2021.

[4] S. Yasunobu, S. Miyamoto, and H. Ihara, "A fuzzy control for train automatic stop control," *Transactions of the society of instrument and control engineers*, vol. 19, no. 11, pp. 873–880, 1983.

[5] M. Erman, A. Mohammed, and E. Rakus-Andersson, "Fuzzy logic applications in wireless communications.," in *IFSA/EUSFLAT Conf.*, pp. 763–767, Citeseer, 2009.

[6] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[7] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy systems*, vol. 3, no. 3, pp. 370–379, 1995.

[8] R. Gray, "Vector quantization," *IEEE Assp Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[9] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1-3, pp. 1–6, 1998.

[10] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[11] H. Chiroma, U. A. Abdullahi, A. A. Alarood, L. A. Gabralla, N. Rana, L. Shuib, I. A. T. Hashem, D. E. Gbenga, A. I. Abubakar, A. M. Zeki, *et al.*, "Progress on artificial neural networks for big data analytics: a survey," *IEEE Access*, vol. 7, pp. 70535–70551, 2018.

[12] C. Sharma, "Big data analytics using neural networks," Master's thesis, The Faculty of the Department of Computer Science, San Jose State University, San Jose, California, Spring, 2014.

[13] M. Oliva Riera, "Quantitative methods for big data: Neural networks," 2018.

[14] V. Ramesh, P. Baskaran, A. Krishnamoorthy, D. Damodaran, and P. Sadasivam, "Back propagation neural network based big data analytics for a stock market challenge," *Communications in Statistics-Theory and Methods*, vol. 48, no. 14, pp. 3622–3642, 2019.

[15] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[16] E. M. Izhikevich, *Dynamical systems in neuroscience*. MIT press, 2007.

[17] D. Carlson and L. Carin, "Continuing progress of spike sorting in the era of big data," *Current opinion in neurobiology*, vol. 55, pp. 90–96, 2019.

[18] J. Vodák, D. Šulyová, and M. Kubina, "Advanced technologies and their use in smart city management," *Sustainability*, vol. 13, no. 10, p. 5746, 2021.

[19] K. Sudarshan, R. R. Hegde, K. Sudarshan, S. Patil, *et al.*, "Smart agriculture monitoring and protection system using iot," *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*, vol. 2, no. 12, pp. 308–310, 2019.

[20] S. Oberlin, "Machine learning, cognition, and big data," *CA Technology Exchange*, vol. 44, p. 112, 2012.

[21] G. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *NIPS*, vol. 15, pp. 833–840, Citeseer, 2002.

[22] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[23] W. Lueks, B. Mokbel, M. Biehl, and B. Hammer, "How to evaluate dimensionality reduction," in *Proceedings of the Workshop-New Challenges in Neural Computation*, vol. 5, pp. 29–37, Citeseer, 2011.

[24] Y. Qian, X. Yin, J. Kong, J. Wang, and W. Gao, "Low-rank graph optimization for multi-view dimensionality reduction," *PloS one*, vol. 14, no. 12, p. e0225987, 2019.

[25] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[26] T. A. Feo and M. G. Resende, "Greedy randomized adaptive search procedures," *Journal of global optimization*, vol. 6, no. 2, pp. 109–133, 1995.

[27] R. A. Gallego, R. Romero, and A. J. Monticelli, "Tabu search algorithm for network synthesis," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 490–495, 2000.

[28] M. Clerc, *Particle swarm optimization*, vol. 93. John Wiley & Sons, 2010.

[29] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.

[30] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial life*, vol. 5, no. 2, pp. 137–172, 1999.

[31] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European journal of operational research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[32] V. K. Ojha, A. Abraham, and V. Snášel, "Aco for continuous function optimization: A performance analysis," in *2014 14th International Conference on Intelligent Systems Design and Applications*, pp. 145–150, IEEE, 2014.

[33] G. Beni, "The concept of cellular robotic system," in *Proceedings IEEE International Symposium on Intelligent Control 1988*, pp. 57–62, IEEE, 1988.

[34] M. A. Kovacina, *Swarm algorithms: simulation and generation.* PhD thesis, Case Western Reserve University, 2006.

[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[36] T. Davidović, "Bee colony optimization part i: The algorithm overview," *Yugoslav Journal of Operations Research*, vol. 25, no. 1, 2016.

[37] A. Rajasekhar, N. Lynn, S. Das, and P. N. Suganthan, "Computing with the collective intelligence of honey bees–a survey," *Swarm and Evolutionary Computation*, vol. 32, pp. 25–48, 2017.

[38] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[39] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (abc) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[40] F. Bennis and R. K. Bhattacharjya, "Nature inspired methods for metaheuristics optimization: algorithms and applications in science and engineering: vol 16," 2020.

[41] J. H. Kim, H. M. Lee, D. Jung, and A. Sadollah, "Performance measures of metaheuristic algorithms," in *Harmony search algorithm*, pp. 11–17, Springer, 2016.

[42] A. E. Charalampakis and G. C. Tsiatas, "Critical evaluation of metaheuristic algorithms for weight minimization of truss structures," *Frontiers in Built Environment*, vol. 5, p. 113, 2019.

[43] M. Nanda and A. Kumar, "Meta-heuristic algorithms for resource allocation in cloud," in *Journal of Physics: Conference Series*, vol. 1969, pp. 01–11, IOP Publishing, 2021.

[44] D. Whitley, S. Rana, J. Dzubera, and K. E. Mathias, "Evaluating evolutionary algorithms," *Artificial intelligence*, vol. 85, no. 1-2, pp. 245–276, 1996.

[45] D. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 1, pp. 215–220, IEEE, 2003.

[46] "Iris data set." Disponible en línea: https://archive.ics.uci.edu/ml/datasets/iris (Fecha de consulta: 23 de Septiembre de 2021).

[47] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[48] S. K. Ghosh, B. Biswas, and A. Ghosh, "A novel approach of retinal image enhancement using pso system and measure of fuzziness," *Procedia Computer Science*, vol. 167, pp. 1300–1311, 2020.

[49] B. Khomri, A. Christodoulidis, L. Djerou, M. C. Babahenini, and F. Cheriet, "Particle swarm optimization method for small retinal vessels detection on multiresolution fundus images," *Journal of biomedical optics*, vol. 23, no. 5, p. 056004, 2018.

[50] K. Aurangzeb, S. Aslam, M. Alhussein, R. A. Naqvi, M. Arsalan, and S. I. Haider, "Contrast enhancement of fundus images by employing modified pso for improving the performance of deep learning models," *IEEE Access*, vol. 9, pp. 47930–47945, 2021.

[51] S. Davies, *Learning in spiking neural networks*. PhD thesis, Citeseer, 2013.

[52] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*. PWS Publishing Co., 1997.

[53] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[55] P. Kim, "Matlab deep learning," *With machine learning, neural networks and artificial intelligence*, vol. 130, no. 21, 2017.

[56] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[57] E. Morales De la Rosa, "Prototipo en fpga de emulador de red neuronal pulsada," Master's thesis, CINVESTAV-IPN, México, 2019.

[58] P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.

[59] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[60] E. M. Izhikevich, *Dynamical systems in neuroscience*. MIT press, 2007.

[61] F. Cicirelli, A. Guerrieri, C. Mastroianni, G. Spezzano, and A. Vinci, *The Internet of Things for smart urban ecosystems*. Springer, 2019.

[62] W. Ejaz and A. Anpalagan, *Internet of things for smart cities: technologies, big data and security*. Springer, 2019.

[63] V. Gazis, "A survey of standards for machine-to-machine and the internet of things," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 482–511, 2016.

[64] "Crecimiento de la población en méxico." Disponible en línea : http://cuentame.inegi.org.mx/poblacion/habitantes.aspx?tema=P (Fecha de consulta: 05 de Octubre de 2021).

[65] W. Stallings, "Ipv6: the new internet protocol," *IEEE Communications Magazine*, vol. 34, no. 7, pp. 96–108, 1996.

[66] E. Adler, "Here's why the "internet of things" will be huge and drive tremendous value for people and businesses'," *Business Insider*, vol. 7, 2013.

[67] G. George, M. Haas, and A. Pentland, "Big data and management," 2014.

[68] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Hung Byers, "Mckinsey global institute, big data: The next frontier for innovation, competition, and productivity." May 2011, Disponible en línea: https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi_big_data_full_report.pdf (accessed on 23 June 2021).

[69] "Power-data, ¿en qué consites el big data?." Disponible en línea: https://www.powerdata.es/big-data (Fecha de consulta: 07 de Octubre de 2021).

[70] "Handwritten digits data set." Disponible en línea: https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits (accessed on 29 April 2021).

[71] "Flowers, fruits and faces data set." Disponible en línea: https://es.dreamstime.com/foto-de-archivo-sistema-de-caras-de-la-gente-image79273852 (accessed on 22 April 2021).

[72] W. Huang, Y. Huang, H. Wang, Y. Liu, and H. J. Shim, "Local binary patterns and superpixel-based multiple kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4550–4563, 2020.

[73] Z. N. L. Ji, "Texture image classification with noise-tolerant local binary pattern," *Journal of Computer Research and Development*, vol. 53, no. 5, p. 1128, 2016.

[74] I. Muslihah and M. Muqorobin, "Texture characteristic of local binary pattern on face recognition with probabilistic linear discriminant analysis," *International Journal of Computer and Information System (IJCIS)*, vol. 1, no. 1, pp. 22–26, 2020.

[75] "Fused optical-radar data set." Disponible en línea: https://archive.ics.uci.edu/ml/datasets/Crop+mapping+using+fused+optical-radar+data+set# (accessed on 29 April 2021).

[76] S. Ji, Z. Wuang, and Y. Wei, "Principal component analysis and autoencoders." Disponible en línea: http://people.tamu.edu/~sji/classes/PCA.pdf (accessed on 22 April 2021).

[77] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[78] "Swiss roll dataset." Disponible en línea: http://people.cs.uchicago.edu/~dinoj/manifold/swissroll.html (Fecha de consulta: 10 de Octubre de 2021).

[79] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, no. 7-9, pp. 1431–1443, 2009.

[80] W. Lueks, B. Mokbel, M. Biehl, and B. Hammer, "How to evaluate dimensionality reduction?-improving the co-ranking matrix," *arXiv preprint arXiv:1110.3917*, 2011.