Universidad Nacional Autónoma de México

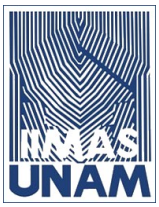
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

EVALUACIÓN DE MAPAS AUTO-ORGANIZADOS Y ALGORITMOS GENÉTICOS PARA APLICACIONES DE CLASIFICACIÓN CON APRENDIZAJE SEMI-SUPERVISADO

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN



PRESENTA:

ALVARO CALLEJAS TAVERA

TUTOR:

ERIK MOLINO MINERO RE

Unidad Académica del IIMAS en Yucatán, Marzo, 2021

A mi familia por su constante apoyo y aliento

A mi compañera de vida y amigos

Al Instituto y a la Universidad, por la formación que me han dado.

Es gracias a ustedes que es posible el presente trabajo.

En verdad, gracias.

Yo.

Reconocimientos

Quisiera agradecer a mi escuela, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, por las experiencias y enseñanzas de vida que me aportaron.

A mi director y amigo Dr. Erik Molino Minero Re por su apoyo, guía y conocimiento aportado dentro y fuera del desarrollo del proyecto de investigación.

A Fátima Yovana Cocom Nah por la ayuda durante la edición de las imágenes y su constante apoyo y aliento durante el proceso.

Al programa UNAM-PAPIIT: IA102918 y IA103420 que apoyó a la investigación realizada.

También quisiera reconocer a CONACYT y al posgrado por brindarme el apoyo económico durante mis estudios de maestría y haberme permitido realizar una movilidad con fines académicos.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Antecedentes

El proceso de clasificación consiste en agrupar y organizar información con base a características que les permita diferenciarse y separar los elementos. Para esto, es necesario conocer la estructura de los datos para identificar y extraer parámetros medibles (características) que ayuden a la separación y creación de grupos del conjunto total. Etiquetar las instancias a menudo es complicado ya que se requiere inversión económica y de tiempo para obtenerlas, dado que se necesita esfuerzo por parte de los anotadores humanos con experiencia para realizar esta tarea.

Con la investigación y desarrollo de técnicas y herramientas de aprendizaje computacional, redes neuronales y algoritmos genéticos, así como la generación diaria de datos en grandes cantidades y la creciente necesidad de procesar, clasificar y agrupar la información, la metodología de clasificar datos ha cambiado con el uso de técnicas automatizadas por medio de algoritmos especializados. Estos han sido desarrollados bajo conceptos diferentes para realizar estas tareas. Por lo tanto, actualmente se encuentran en la literatura gran variedad de opciones para resolver este tipo de problemas siguiendo enfoques diferentes con resultados interesantes.

El aprendizaje semi-supervisado surge de los paradigmas en los que se divide el aprendizaje computacional: supervisado, no supervisado y por refuerzo. Estas técnicas de aprendizaje emplean datos de entrenamiento etiquetados y no etiquetados para

clasificar información, en este enfoque es común tener una pequeña cantidad de datos etiquetados junto a un volumen grande de datos no etiquetados. Al combinar ambos tipos de datos se ha observado que se mejora la exactitud del aprendizaje en los algoritmos. Además, esta forma especial de clasificación en donde se usan grandes cantidades de datos no etiquetados, junto con pocos datos etiquetados construyen mejores clasificadores [1].

1.2. Planteamiento del problema: sísmica de reflexión

La sísmica de reflexión es un método de exploración geofísica utilizado para estimar propiedades del subsuelo midiendo y estudiando ondas acústicas que se propagan y se reflejan en los distintos estratos [2]. Esta información obtenida en campo es procesada y transformada en trazas sísmicas, que a su vez se unen para formar una imagen de sísmica de reflexión, donde los picos de amplitud representan reflejos y permiten identificar estructuras geológicas. Este tipo de información se ha estudiado ampliamente en la literatura, sin embargo, un campo de investigación que ha comenzado a ser muy activo es el de utilizar técnicas de aprendizaje computacional para identificar regiones de la imagen sísmica donde la firma de las ondas tienen características similares [2] [3]. No obstante, conocer la estructura de los materiales que conforman el área analizada es complejo, puesto que el subsuelo es un medio heterogéneo, y la información sísmica no es suficiente para resolver completamente la litología, esto es, definir etiquetas en los datos. Sin embargo, la información sísmica es muy importante para iniciar el proceso de clasificación, y por ello resulta importante lograr agrupar de manera eficiente regiones donde los reflejos sísmicos son similares, para generar grupos iniciales y posteriormente clasificarlos por zonas homogéneas.

Por ello, en este trabajo de tesis se aborda el problema de agrupamiento de datos donde no están claramente definidas las diferencias entre los grupos. En el caso de los datos sísmicos, se busca poder identificar regiones similares asociadas a las distintas capas del subsuelo, desde el punto de vista del aprendizaje no supervisado, en donde

la estructura de la información no es conocida en su totalidad, por lo que la tarea de agrupamiento resulta ser compleja. Sin embargo, el alcance de las ideas desarrolladas en este trabajo no está limitado solo a este tipo de datos.

El desarrollo de este trabajo cuenta con una base de datos compuesta por trazas sísmicas de reflexión de un cubo sísmico, con el cual se pueden construir imágenes sísmicas, pero donde no se tiene información de la litología (etiquetas) de la región que se está analizando, por lo que la estructura es desconocida. Por ello, se desarrolla una metodología de trabajo para implementar herramientas de manejo de grandes volúmenes de información de forma eficiente y manejar los cubos sísmicos que están compuestos por una gran cantidad de información que necesita procesarse y analizarse.

1.3. Objetivo

Definición de metodologías de entrenamiento para la creación de grupos y clasificación de datos empleando mapas auto-organizados y algoritmos genéticos. Así como la evaluación de estos métodos empleando diferentes bases de datos etiquetadas y no etiquetadas con distintas estructuras de datos.

1.4. Contribución y relevancia

El problema de investigación surgió de la necesidad de agrupar y clasificar la información sísmica en secciones con características sísmicas similares por medio de algoritmos especializados de aprendizaje máquina. Para esto se crearon herramientas computacionales de uso general, es decir, que pueden ser aplicadas a cualquier base de datos que cumpla con los requerimientos mínimos de pre-procesamiento sin importar su procedencia y su estructura. La importancia de contar con herramientas flexibles de propósito general que solucionen problemas con pocos ajustes, permite utilizarlas con una amplia variedad de problemas, por lo que su campo de aplicación es muy amplio.

Además, se aborda el problema de agrupamiento de datos de forma general con dos diferentes técnicas: algoritmos genéticos y redes neuronales de tipo mapas auto-

organizados. En la primera técnica la clave es definir bien la función objetivo del problema, así como los operadores genéticos a utilizar, tipo de codificación y métodos de selección. Mientras tanto, con los mapas auto-organizados se necesita entrenarlos y dividirlos en grupos con base en una metodología de agrupamiento de neuronas implementada, generando clases en el mapa de neuronas utilizado para clasificar la información. El modelo de la red neuronal que fue evaluado con el uso de distintas bases de datos etiquetadas y fue probado con diferentes tipos de datos.

Uno de los problemas sin resolver del aprendizaje no supervisado es conocer la cantidad de grupos que pueden ser formados con los datos, por lo que existen diferentes alternativas que miden la calidad de los grupos generados sin importar la herramienta utilizada para crearlos. Por lo anterior, se implementó una metodología de índices para medir la calidad en las clases formadas. Los resultados se compararon con las técnicas existentes en la literatura para seleccionar la cantidad de grupos en los que se dividieron las bases de datos. Por otra parte, implementar herramientas de visualización de resultados es necesario e importante debido a que la mayoría de los problemas de clasificación tienen bases de datos de múltiples dimensiones. Por lo tanto, contar con este tipo de herramientas permite dar una interpretación gráfica y lógica al trabajo desarrollado, visualizando los resultados en una imagen o gráfica.

1.5. Metas

- Recopilación de información del estado del arte sobre técnicas de aprendizaje
- Implementación de algoritmos para la agrupación de datos con diferentes bases de prueba
- Pre-procesamiento de los datos antes de iniciar el proceso de entrenamiento y agrupamiento de datos
- Planteamiento del problema como función objetivo (codificación) en el proceso de optimización del algoritmo genético

- Implementación de algoritmos de agrupamiento basados en algoritmos genéticos y en mapas auto-organizados
- Evaluación de la calidad de los grupos creados por medio de la implementación de técnicas de evaluación por índices
- Evaluación de la metodología desarrollada haciendo uso de diferentes bases de datos etiquetadas
- Creación de grupos (etiquetas) y clasificación de datos con la base de datos sísmicos no etiquetada

1.6. Estructura de la tesis

Este trabajo de investigación está dividido en cinco capítulos. El primer capítulo es la introducción, en donde se abordan los antecedentes de la investigación, planteando el problema que dio origen a la investigación, así como el objetivo, metas y la relevancia como las contribuciones y técnicas desarrolladas.

En el segundo capítulo se abordan los enfoques del aprendizaje máquina y sus divisiones. Después se explican algunas técnicas de pre-procesamiento de datos para generar los vectores de entrenamiento. Seguidamente se desarrolla la teoría de los algoritmos utilizados para agrupar datos: algoritmos genéticos y mapas auto-organizados. Además, se presentan tres metodologías para evaluar la calidad del agrupamiento en los datos y por último, se presentan las métricas de evaluación para clasificación de datos.

En el tercer capítulo se desarrolla la metodología utilizada para el proyecto de investigación. Iniciando se muestra la codificación del problema y planteamiento de la función objetivo para agrupar los datos mediante el algoritmo genético. Así mismo, se describe la metodología de tres etapas para agrupar los mapas entrenados de las redes SOM. También se presentan las herramientas de visualización y funcionalidad implementadas para la clasificación y se explica el pre-procesamiento hecho a las base de datos sísmicos para generar los vectores de entrenamiento. Por último, se detallan las

bases de datos utilizadas en las pruebas y la metodología que se utilizó para realizarlas.

En el cuarto capítulo se muestran los resultados después de la evaluación de las bases de datos etiquetadas de prueba, así como el resultado de la clasificación de los datos sísmicos. Además, se muestran los grupos formados con el modelo de red neuronal y se visualizan los datos empleando técnicas de visualización de datos de alta dimensión.

Por último, en el capítulo 5 se muestran las conclusiones del trabajo después de comparar los resultados con el objetivo y las metas planteadas, así como el trabajo a futuro con las distintas ideas que han surgido como resultado de este trabajo. Además, se proponen mejoras a la metodología desarrollada.

Capítulo 2

Aprendizaje computacional para el análisis y clasificación de datos

2.1. Paradigmas del aprendizaje computacional

El aprendizaje máquina surgió como subcampo de las ciencias de la computación donde se combinan conceptos de inteligencia artificial para desarrollar técnicas para las computadoras [4]. Los modelos aprenden a partir de heurísticas o procesos de optimización, siendo capaces de generalizar comportamientos para un conjunto más extenso.

El aprendizaje maquina puede ser divido en tres enfoques de aprendizaje: supervisado, no supervisado y por refuerzo. Los tres paradigmas están conformados por diferentes técnicas computacionales que resuelven distintos problemas con los datos analizados. Por otra parte, existe otro enfoque de aprendizaje basado en los dos primeros mencionados, denominado aprendizaje semi-supervisado. En la Figura 2.1 se muestra la división en ramas del aprendizaje máquina según el problema que se desea resolver.

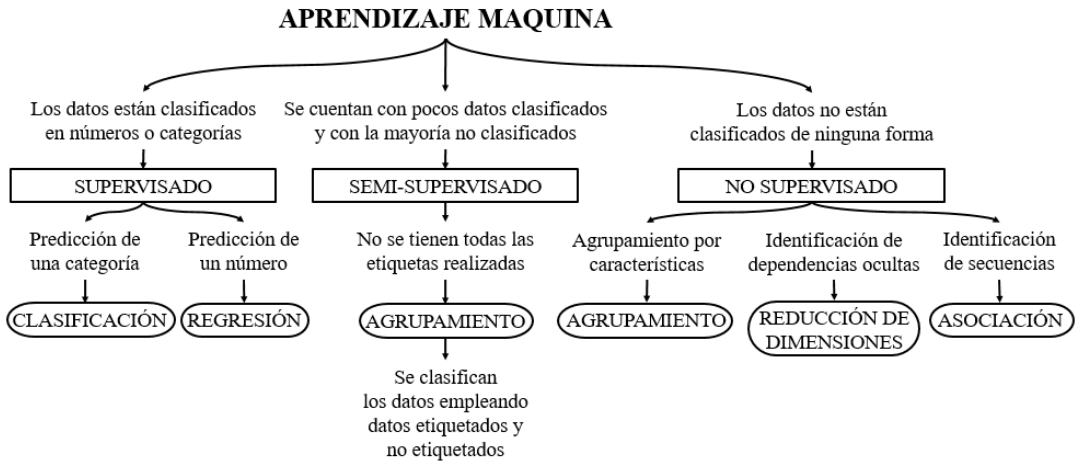


Figura 2.1: Estrategias del aprendizaje máquina.

2.1.1. Aprendizaje supervisado

El aprendizaje supervisado es un área del aprendizaje computacional, donde el objetivo es aprender a mapear de x a y dado un conjunto de entrenamiento compuesto de (x_i, y_i), donde las y_i son las etiquetas u objetivos y las x_i son los ejemplos [5]. Este mapeo es realizado por medio de una función matemática con capacidad de estimar la clase correspondiente a cualquier dato de entrada después de haber realizado el entrenamiento, generalizando situaciones a partir del conocimiento adquirido. Cuando las etiquetas son continuas, es decir, $y = R^d$ la tarea de mapeado es llamada regresión. Existen dos familias de algoritmos para el aprendizaje supervisado: modelos generativos y algoritmos discriminatorios. En los modelos generativos se trata de modelar la densidad de clase conjunta $p(x|y)$ [5], ésta puede ser inferida como se muestra en 2.1.

$$p(y_i|x) = \frac{p(x|y_i)p(y_i)}{\sum_{k=1}^{n} p(x|y_k)p(y_k)} \tag{2.1}$$

Por otra parte, los algoritmos discriminatorios no estiman cómo son los datos x_i que han sido generados, sino más bien estiman la probabilidad conjunta $p(y|x)$ de las etiquetas y los ejemplos dados.

2.1.2. Aprendizaje no supervisado

El aprendizaje no supervisado es un área del aprendizaje de máquina en donde el objetivo es encontrar estructuras dentro de los datos, partiendo de que no existe un conocimiento a priori de ellos. A diferencia del aprendizaje supervisado los modelos no supervisados no requieren de etiquetas, las cuales pueden o no existir, por lo que los modelos se ajustan al conjunto de variables de entrada [6].

Existen otras técnicas no supervisadas como la estimación de cuantiles, agrupamiento, detección de anomalías y reducción de la dimensionalidad. Dentro de estas técnicas el análisis del agrupamiento es utilizado para crear grupos y segmentar las bases de datos con características compartidas para inferir las relaciones entre los datos [7].

2.1.3. Aprendizaje semi-supervisado

El aprendizaje semi-supervisado es una técnica intermedia entre el aprendizaje supervisado y no supervisado, que emplea datos de entrenamiento etiquetados y no etiquetados. La idea del método semi-supervisado es que la parte de la base de datos etiquetada sea proporcionalmente pequeña con respecto a la base de datos total, además de que la complejidad del problema sea alta por lo que hacer un entrenamiento supervisado no sea suficiente. Por ello se explora esta opción semi-supervisada, para aumentar el número de etiquetas de manera adecuada, tratando de minimizar los errores de clasificación.

El objetivo del aprendizaje semi-supervisado es similar al del aprendizaje supervisado, es decir, estimar un valor de salida para un valor x_i de entrada. Sin embargo, si el número de grupos y la estructura de las clases son desconocidas pueden ser inferidas por medio de los datos. Además, se ha observado que la información no etiquetada mejora considerablemente la exactitud durante el proceso de aprendizaje [5].

Esta técnica es muy útil cuando la adquisición de los datos es costosa, tardada o es un nuevo experimento ya que con poca información etiquetada se puede llevar a cabo una inferencia sobre los demás datos empleando de forma conjunta datos etiquetados

y no etiquetados. Un enfoque alterno del aprendizaje semi-supervisado es modelar la distribución conjunta $p(x|y)$ de las características y las etiquetas, tratando a los datos no etiquetados como datos faltantes del modelo [5]. Una de las técnicas más comunes para maximizar esa similitud bajo este enfoque es el algoritmo esperanza-maximización [8].

2.2. Técnicas de pre-procesamiento de datos

Una de las etapas más importantes del aprendizaje de máquina es el pre-procesamiento de datos, se utiliza para revisar las características de los datos y el tipo de distribución que tienen, además evita que un subconjunto de datos predomine sobre otros (con la normalización o estandarización), permite revisar si hay valores faltantes (y se puede decidir qué hacer al respecto), entre otros.

Si el pre-procesamiento no se hace de forma acertada, se corre el riesgo de que los resultados no representen adecuadamente el proceso que se desea estudiar. Por lo tanto, la forma en como se pre-procesan los datos y se ingresan al modelo es lo más importante antes comenzar con el análisis [9]. Así también, cuando las bases de datos son construidas, es importante invertir tiempo en la limpieza de los datos para evitar problemas futuros durante el entrenamiento. Las transformaciones realizadas a los datos no deben influir en la naturaleza del proceso que describen estos [9]. Las técnicas de pre-procesamiento ayudan en diversos problemas de clasificación siendo herramientas versátiles que pueden ser clasificadas en grupos, a continuación se listan algunas técnicas utilizadas en el trabajo de investigación:

- Aseguramiento de la calidad en los datos
- Agregación de características
- Reducción de la dimensionalidad
- Estandarización
- Normalización

Cuando la información es capturada para crear las bases de datos, existen problemas tales como el formato de captura de los datos, lo que genera que la información sea inconsistente y no confiable. Estos aspectos inherentes al proceso de construir la base de datos son inevitables ya que se debe a diversos factores como el error humano, limitaciones en los dispositivos de medición, entre otros. Por lo tanto, es importante asegurar la calidad de los datos antes de trabajar con ellos.

Entre los problemas más comunes se encuentran: datos faltantes, inconsistencia de valores y valores duplicados. Los datos faltantes pueden ser corregidos eliminando el ejemplo con datos faltantes o con técnicas como la imputación de información [10]. Ahora bien, si los datos son duplicados estos deben ser removidos para que el proceso de entrenamiento no se vea influenciado. Por otra parte, cuando se cuenta con gran cantidad de información y se quiere tener una mejor perspectiva del comportamiento del modelo se extraen características de los datos, como por ejemplo los momentos estadísticos (media, desviación estándar, entre otros).

La mayoría de los problemas están descritos mediante bases de datos con muchos atributos (dimensiones), los cuales pueden ser visualizados mediante técnicas de visualización de datos de alta dimensionalidad. Además, a mayor número de dimensiones mayor complejidad en la base de datos, por lo que aveces es conveniente utilizar técnicas que reduzcan la dimensionalidad. El proceso se lleva a cabo mediante un mapeo del espacio original a un nuevo espacio de menor dimensión, una de las técnicas más conocidas es el análisis de componentes principales o PCA *(Principal Component Analysis)* el cual expresa un conjunto de variables en un conjunto de combinaciones lineales de factores no correlacionados entre sí, creando nuevos espacios y características (agregación de características) que son combinaciones lineales de los datos anteriores [11]. Entre los beneficios de utilizar este tipo de técnicas está un mejor análisis de la información, visualización de los datos, facilidad en el despliegue de resultados y modelos menos complejos.

La estandarización de una base de datos es una práctica muy común en el aprendizaje de máquina, una forma de realizarla es removiendo la media y escalando los datos

a una varianza unitaria como se muestra en la ecuación 2.2.

$$Z = \frac{x - \mu}{\sigma} \tag{2.2}$$

Lo que genera que los datos sean escalados y centrados independientemente de cada característica, esto ayuda a los modelos que están basados en la distribución de atributos como los procesos Gausianos. Por otra parte, la normalización de datos se refiere al escalamiento de atributos numéricos en rangos como 0 a 1 o -1 a 1 entre otros. Esta técnica es útil para escalar los atributos de entrada para un modelo basado en la magnitud de los valores, en la ecuación 2.3 se muestra una estandarización por el mínimo y máximo escalando a los valores entre 0 a 1.

$$X_{esc} = \frac{x - \min x}{\max x - \min x} \tag{2.3}$$

2.3. Algoritmos genéticos

Los Algoritmos Genéticos (AGs) son métodos adaptativos utilizados en problemas de búsqueda y optimización. Están basados en el proceso genético y evolutivo de los organismos que comienzan a mostrar características acordes a la selección natural y la supervivencia con el transcurrir de generaciones. Estos fundamentos biológicos dan pie a los algoritmos genéticos establecidos en 1975 por Holland [12], los cuales encuentran soluciones para problemas que se presentan proponiendo una diversidad de soluciones que se aproximan a los valores óptimos para resolver el problema dependiendo de igual forma de los parámetros de configuración.

En la naturaleza la competencia entre individuos de una población por recursos como alimento, refugio y agua ocurre de manera natural y cíclica, incluso este comportamiento se extiende a individuos de una misma especie cuando compiten por un individuo de su especie para la reproducción. Los individuos más exitosos en sobrevivir y en atraer a otros individuos cuentan con una mayor probabilidad de procrear descendientes para las futuras generaciones. Sin embargo, los individuos menos adaptados

cuentan con una probabilidad menor de generar descendientes y por ende la pérdida de su material genético en las próximas generaciones.

El resultado de la recombinación de las características provenientes de diferentes ancestros a veces produce descendientes con una muy alta capacidad de adaptación, por lo que las especies que evolucionan de estos "súper individuos" logran características mejores adaptadas al entorno en donde habitan. Los algoritmos genéticos se basan en estos principios de interacción que ocurre en la naturaleza mediante las poblaciones de individuos, en donde cada individuo representa una solución probable a un problema. A cada uno le es asignado una calificación o puntuación relacionado a la efectividad de la solución que plantea, es decir, se valora el grado de efectividad de un organismo para competir por los recursos existentes [13]. Cuanto mayor es la puntuación mayor es la probabilidad de que el individuo pueda ser seleccionado para reproducirse, combinando su material genético con otro seleccionado de la misma forma.

La cruza de los individuos producirá nuevos descendientes que compartirán características de sus padres preservadas al menos hasta la siguiente generación. Ésta nueva población de soluciones contiene una mayor diversidad en las características adquiridas, ya que es el resultado de la recombinación de la generación anterior. Estas características son propagadas a través de generaciones mejorando a los individuos para que la población vaya convergiendo a una solución.

A pesar de que las mejoras en las características favorecen la velocidad de convergencia a la solución óptima, existen técnicas complementarias para mejorar la tasa de convergencia. El elitismo consiste en seleccionar la mejor solución durante la generación actual y al final del proceso de mutación reemplazar la peor solución por la mejor, en algunas ocasiones el reemplazo es doble, es decir, se sustituyen las últimas dos soluciones con base en el vector de aptitud obtenido. Si bien este proceso ayuda a acelerar la convergencia hacia la solución óptima, también puede hacer que el algoritmo quede atrapado en un mínimo local, por lo que es importante mantener un equilibrio entre los parámetros de configuración y las técnicas complementarias.

Entonces los algoritmos genéticos descritos por Goldberg [14] y Michalewicz [15]

evolucionan poblaciones de individuos (cromosomas) mediante procesos aleatorios imitando a la naturaleza por medio de operaciones de recombinación y mutación que ocurren en los procesos evolutivos. Todo esto basado en la función objetivo desarrollada para resolver el problema de optimización, seleccionando a las mejores soluciones con el paso de las siguientes generaciones [15], en la Figura 2.2 se observa el ciclo evolutivo del algoritmo genético.

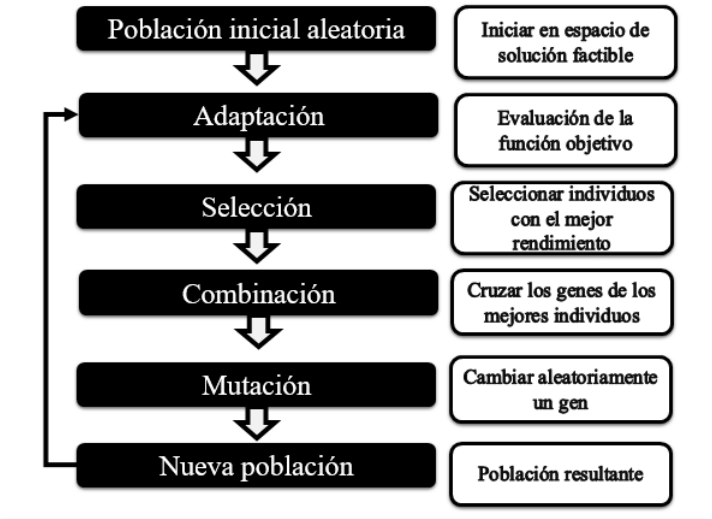


Figura 2.2: Ciclo evolutivo del algoritmo genético.

2.3.1. Tipos de codificación

Las posibles soluciones al problema (individuos) son representados como una secuencia de genes que conforman una cadena de valores conocidas como cromosomas. El alfabeto utilizado para representar a los individuos no necesariamente debe estar constituido por valores 0 ó 1, aunque buena parte de la teoría desarrollada en la que se fundamentan los algoritmos genéticos emplean dicho alfabeto [12].

El alfabeto utilizado para representar a los individuos en la población se denomina codificación, en algoritmos genéticos es la representación del genotipo de cada individuo, que en términos biológicos es el conjunto de parámetros de un cromosoma que contiene la información necesaria para constituir a un organismo. Esta codificación puede tener

el alfabeto binario, entero o real [16], en la Figura 2.3 se observan algunas posibles decodificaciones del genotipo.

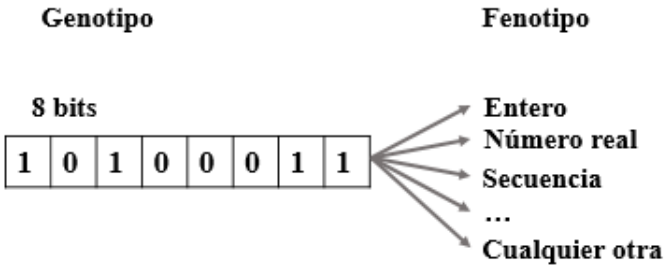


Figura 2.3: Codificación binaria del genotipo en AGs.

En la Figura 2.3 se observa la codificación del problema en binario, no obstante existen otras alternativas que deben ser elegidas considerando el problema, ya que este factor puede afectar al desempeño del algoritmo en la velocidad de convergencia a la solución. La codificación binaria al utilizar el alfabeto 0,1 puede ser interpretada como activación y desactivación de parámetros o inclusive como secuencias de números codificados en intervalos de longitud constantes en los genes dentro del cromosoma. Si se requiere hacer operaciones con el fenotipo basado en otro sistema numérico es necesario decodificar al genotipo para poder expresar el contenido del cromosoma en un valor, en la ecuación 2.4 se observa una forma de decodificación de binario a decimal, en donde a y b representan los rangos mínimo y máximo calculados, después de seleccionar la precisión en decimales deseada y la cantidad de m bits necesarios para conseguir esa precisión.

$$x = a + (decimal(100...11_2)(\frac{b-a}{2^m - 1}) \tag{2.4}$$

Por otra parte, cuando se codifica el problema de forma real se elimina el mapeo genotipo-fenotipo, es decir, se elimina la decodificación ya que en cada posición del cromosoma (locus), se encuentra el número real que representa una posible solución por

medio de la evaluación de la función objetivo, lo que reduce el tiempo de procesamiento y los cálculos realizados por los operadores genéticos. Si en cambio el problema está planteado en forma de permutaciones como por ejemplo: el problema del agente viajero [17], los conjuntos de valores enteros expresados en cada gen representan al conjunto n de elementos únicos que conforman el problema. De igual forma se elimina el mapeo genotipo-fenotipo en esta codificación y sus operadores pueden ser con repetición o sin repetición del valor entero en otra posición. En general no existe una regla sobre la forma de codificar el problema, no obstante se recomienda analizar bien el tipo de problema que se desea resolver y con base al objetivo que se desea alcanzar, elegir la mejor forma de codificar la solución evitando mayor complejidad ya que la velocidad de convergencia también depende de la forma en como las soluciones están codificadas.

2.3.2. Métodos de selección

Antes de la fase reproductiva, los individuos son seleccionados entre la población para cruzarse y producir a los nuevos individuos (descendientes) que después del operador de mutación conformarán la siguiente generación. Para seleccionar a los padres de los descendientes existen diversos métodos que utilizan el azar basado en la conservación del mejor individuo después de su evaluación en la función objetivo. A continuación se listan los métodos de selección más conocidos.

2.3.2.1. Selección por ruleta

La selección por ruleta, también conocida como el muestreo estocástico con remplazo, es el método de selección más usado. Esta técnica consiste en asignar un segmento proporcional de la ruleta a los individuos con base en la evaluación del desempeño individual y general de la población, para que sea seleccionada la sección de la ruleta por medio del azar. Los individuos mejor evaluados, es decir, con el mejor desempeño en la evaluación de la función objetivo son los que cuentan con mayor probabilidad de ser elegidos. De forma análoga el proceso se visualiza como una ruleta en la que cada una de las porciones representa a un individuo de forma proporcional a su desempeño, en

la Figura 2.4 se muestra representada la división proporcional.

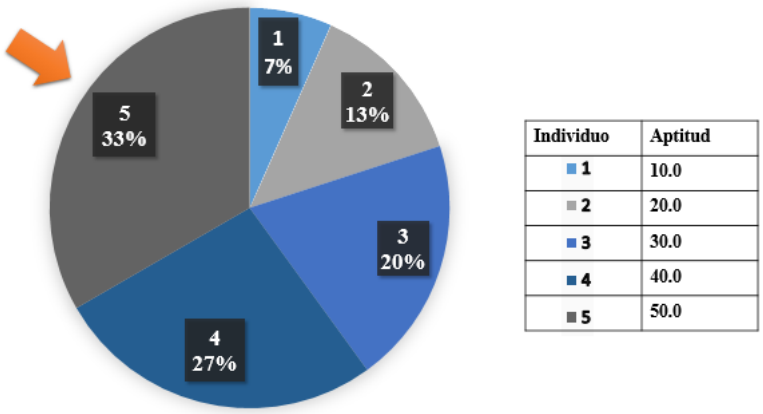


Figura 2.4: Representación del método de selección por ruleta.

Este proceso tiene que ser modificado dependiendo si se busca minimizar o maximizar ya que las proporciones se invierten cuando se desea minimizar la función objetivo, debido a que los individuos con menor valor de desempeño son los que obtienen una proporción mayor. En la Figura 2.5 se muestra la implementación de está metodología en pseudocódigo.

Calcular el valor objetivo $f(x_i)$ para cada cromosoma x_i
Calcular el valor objetivo total para la población:
$F = \sum_{i=1}^{N} f(x_i), i = 1, 2, ..., N$ donde N = tamaño población
Calcular la probabilidad de selección p_i para cada cromosoma x_i:
$p_i = \frac{f(x_i)}{F}, i = 1, 2, ..., N$
Calcular la probabilidad acumulada q_i para cada cromosoma x_i:
$q_i = \sum_{i=1}^{l=1} p_l, i = 1, 2, ..., N$
for 1:N **do**
    Generar un número al azar p en un rango [0,1]
    Escoger el i-ésimo cromosoma x_i tal que $q_{i-1} < p \leq q_i$
end for

Figura 2.5: Pseudocódigo del método de selección por ruleta.

2.3.2.2. Muestreo estocástico universal

El método de muestreo estocástico universal o *SUS* por su siglas en inglés, es una modificación del método por ruleta en donde el muestreo que se implementa es de una sola fase. Esta técnica corrige algunos problemas que presenta el método de selección por ruleta. Dado un conjunto de n individuos y sus valores de aptitud asociados, el algoritmo *SUS* los ordena en una ruleta en donde el tamaño de los cortes asignados a cada uno es proporcional al valor de aptitud. Después, una segunda ruleta es marcada con y marcadores igualmente separados entre si, donde y es el número de selecciones que se desean realizar, si l marcadores caen sobre el mismo individuo, éste es seleccionado l veces. Esto garantiza que ningún individuo sea seleccionado ni más ni menos veces que las esperadas.

2.3.2.3. Torneo

El método de selección por torneo tiene diferentes variantes de aplicación, sin embargo, el método de dos etapas consiste en seleccionar dos individuos de la población aleatoriamente para comparar su desempeño mediante la evaluación de su función objetivo, el que mejor desempeño obtenga pasa a la segunda fase. Seguidamente se selecciona de forma aleatoria a otro individuo de la población y se repite el paso anterior, el ganador pasa a formar parte de la nueva población a la cual se le aplicarán de nueva cuenta los operadores genéticos. Este proceso es iterativo y se repite hasta generar otra población de igual tamaño, debido a que durante el proceso evolutivo el tamaño de la población debe mantenerse constante, en la Figura 2.6 se observa el proceso de selección.

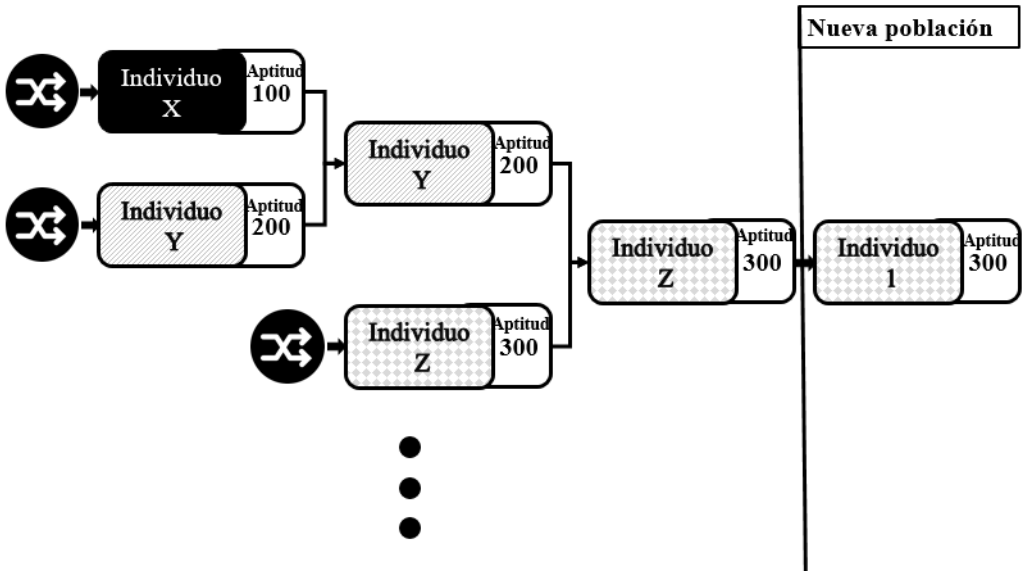


Figura 2.6: Método de selección por torneo en dos fases.

2.3.3. Operadores genéticos

Durante el proceso de la generación de nuevos individuos para encontrar la solución óptima, la reproducción o cruza es el operador genético principal ya que a través de él se generan nuevos individuos mejor adaptados, partiendo de las soluciones actuales. Este operador funciona sobre dos individuos (cromosomas) durante el proceso y genera descendencia recombinando las características de ambos.

La cruza más común se lleva a cabo seleccionando dos padres de la población y cortando sus cadenas de cromosomas en un punto elegido al azar para producir dos sub-cadenas a partir de ese punto de corte. Después se intercambian la información contenida en las sub-cadenas, produciéndose dos nuevos cromosomas descendientes, ambos heredando genes de cada uno de los padres, este operador se conoce como operador de cruza en un punto. Si este mismo concepto es extrapolado con más puntos de corte en donde la información es mezclada intercalando las sub-cadenas, se genera un operador que se conoce como operador de cruza multipunto, en la Figura 2.7 se observa el funcionamiento de ambos operadores.

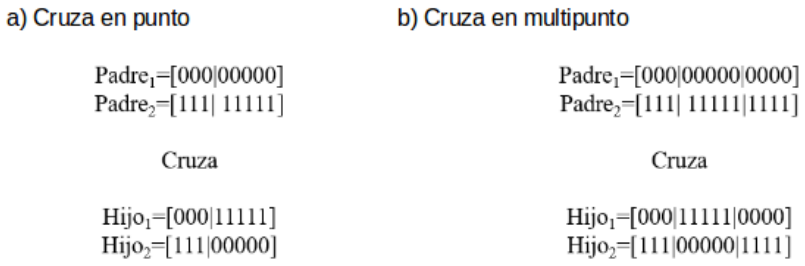


Figura 2.7: a) Operador de cruza en un punto b) Operador de cruza multipunto.

Si la codificación es real el operador de cruza se convierte en una combinación lineal aplicada a cada cromosoma, eligiendo de forma aleatoria a los padres para formar a los nuevos descendientes. Como se observan en las ecuaciones 2.5 y 2.6.

$$h_1 = P_1 + \alpha(P_2 - P_1), \quad donde \quad \alpha[-0.25, 1.25] \tag{2.5}$$

$$h_2 = P_2 + \alpha(P_1 - P_2), \quad donde \quad \alpha[-0.25, 1.25] \tag{2.6}$$

Para el operador de mutación existen diferentes técnicas, entre ellas la más conocida se denomina como mutación por intercambio y es realizada eligiendo a un gen del cromosoma y verificando si la probabilidad de mutación se cumple. Se eligen dos genes aleatoriamente y se los intercambia en el mismo cromosoma, esta operación permite diversificar a la población mediante la exploración en su espacio de búsqueda de soluciones [16], en la Figura 2.8 se observa un ejemplo de mutación por intercambio.

Figura 2.8: Operador de mutación por intercambio.

Si la codificación es real, el proceso de mutación es realizado mediante combinaciones lineales de forma similar que la cruza, en las ecuaciones 2.7 y 2.8 se observa la creación de los nuevos individuos que son formados mediante estos procesos aleatorios.

$$h_i = P_i + rango_i * \delta, \quad donde \quad rango = 0.5 \tag{2.7}$$

$$\delta = \sum_{i=0}^{m-1} \alpha_i * 2^{-i}, \quad donde \quad \alpha[0,1] \quad m = 16 \tag{2.8}$$

2.4. Mapas auto-organizados

El mapa auto-organizado, es un un tipo de red neuronal no supervisada y no paramétrica que es entrenada para producir representaciones discretas del espacio de entrada en forma de mapa. Ésta técnica es muy utilizada durante la fase exploratoria, ya que permite la visualización y análisis de datos multidimensionales [18]. Los vectores de entrada son proyectados en unidades de baja dimensión (neuronas) sobre un mapa que es utilizado para visualizar y explorar las propiedades de la información [19].

Una de las propiedades más importantes de los mapas auto-organizados es la utilización de una función de vecindad para la preservación de la topología del espacio de entrada, puesto que la información es mapeada en base a un mapa organizado usualmente de dos dimensiones, que ayuda a visualizar la estructura de datos en alta dimensión en comparación a otro tipo de redes neuronales. El mapa auto-organizado puede operar en las dos formas como operan las redes neuronales: entrenamiento y mapeo. Para el entrenamiento el mapa es construido usando datos de entrada, mientras que durante el mapeo con el mapa entrenado se agrupan nuevos datos.

El mapa generado está compuesto de unidades ordenadas en una malla compuesta por neuronas, las cuales están asociadas a un vector de pesos de la misma dimensión que los vectores de entrada. La configuración del mapa usualmente consta de un espacio regular de dos dimensiones, en una malla hexagonal o rectangular. Para encontrar un vector multidimensional de datos mapeado basta con evaluar la distancia que puede

ser calculada de varias formas entre las neuronas y los datos seleccionando al de menor distancia, en la Figura 2.9 se observa la estructura de mapa auto-organizado.

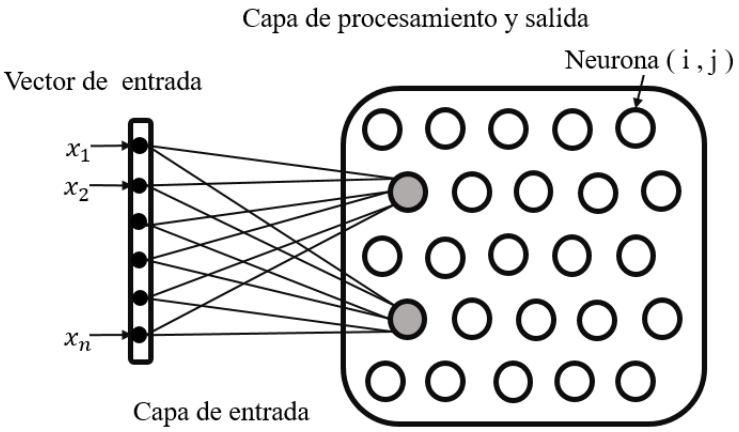


Figura 2.9: Representación del mapa auto-organizado.

Además existen otras configuraciones de la malla, como por ejemplo el uso de mallas toroidales que presentan mayor conectividad entre sus neuronas. Así mismo, otra de las propiedades intrínsecas del mapa es el tamaño, el cual influye en el comportamiento ya que si el mapa es pequeño se comporta de forma similar a K-medias [20] agrupando datos, mientras que para mapas más grandes los datos son mapeados distribuyéndose en el mapa y conservando su topología, en donde también se pueden crear grupos uniendo neuronas mediante una metodología establecida [21].

2.4.1. Algoritmo de los mapas auto-organizados

El proceso de entrenamiento de los mapas auto-organizados se lleva a cabo para que diferentes secciones del mapa respondan de forma similar a ciertos patrones de entrada, por medio del aprendizaje estos patrones son mapeados en los pesos de las neuronas que conforman al mapa. Durante el proceso de aprendizaje la red neuronal funciona como una red elástica que cubre la nube de datos del espacio de entrada, acercándolos unos a otros para mapearlos en las neuronas, ésta característica ayuda a conservar la topología de los datos de entrada. La idea detrás de esta técnica está basada en el manejo de

partes separadas de la corteza cerebral del cerebro humano de la información sensorial, visual y auditiva [22].

El mapa auto-organizado usualmente se representa de forma bidimensional con una malla de neuronas ordenadas y con una distancia vectorial a neuronas vecinas manteniendo una relación de vecindad. Cada neurona i es representada por un vector prototipo $m_i = [m_{i1}, \ldots, m_{i,d}]$ donde d es la dimensión del vector de entrada. El número de neuronas por cada mapa determina la capacidad de generalización y precisión de éste, por lo tanto, es necesario introducir un gran número de datos a la red neuronal que representen lo mejor posible la naturaleza de los vectores de entrada durante el mapeo, llevando a cabo este proceso de forma iterativa [23].

El entrenamiento del mapa auto-organizado es un proceso iterativo pero antes de iniciar el ciclo, los pesos contenidos en las neuronas son inicializados con un pequeño valor aleatorio o muestreado de forma uniforme. Al aplicar la herramienta de análisis de componentes principales *PCA* a los datos de entrada, este muestreo no aleatorio acelera el proceso de aprendizaje puesto que al inicio se tiene una aproximación de los pesos reales. Después de inicializar la matriz de pesos, los vectores de entrenamiento x_i son elegidos de forma aleatoria de la base de datos de entrada. Posteriormente, se calculan las distancias entre el vector x y todas las neuronas para identificar al mejor emparejamiento, que se denomina la *BMU* por sus siglas en inglés (*Best Matching Unit*), es decir, la neurona mapeada más cercana a x como se muestra en 2.9.

$$BMU = c = \min_i ||x - m_i|| \tag{2.9}$$

Después, la matriz de neuronas es actualizada, generando una interacción de acercamiento entre la BMU y sus vecinos topológicos hacia el vector de entrada, siguiendo la regla de actualización que se muestra en la ecuación 2.10.

$$m_i(t+1) = m_i(t) + \alpha(t) h_{ci}(t)[x - m_i(t)] \tag{2.10}$$

donde

$$t = \ tiempo \quad T = \ épocas$$

$$\alpha(t) = \ coeficiente\, de\, aprendizaje:$$

$$\alpha(t) = \alpha_0(t) \exp\left(-\frac{t}{T}\right)$$

$$h_{ci}(t) = \ vecindario\, centrado\, en\, BMU:$$

$$h_{ci}(t) = exp(-\frac{||r_c - r_i||^2}{2\delta^2(t)})$$

$$\delta(t) = \delta_0(t) exp((-\frac{t}{T}))$$

En donde r_c y r_i son las posiciones de las neuronas BMU en la malla del mapa y además α_0 y δ_0 son los parámetros de configuración inicial. Ambas $\alpha(t)$ y $\sigma(t)$ decaen monótonamente con el tiempo de forma exponencial después del paso de actualización de la matriz de pesos. Este proceso es repetido iterativamente actualizando la matriz de pesos por cada vector x durante las épocas seleccionadas, no obstante existe otro método de actualización por lotes en donde el coeficiente de aprendizaje no es utilizado [18]. En la Figura 2.10 se muestra el pseudocódigo del algoritmo del mapa auto-organizado.

Entrada:
Vectores de entrenamiento $X = (x_1, x_2, ..., x_T) \in R^{Txn}$ con T ejemplos de n dimensiones.
Salida:
La matriz de pesos $m = \{m_k | k \in A\}$ entrenada
Configuración:
Configure el coeficiente de aprendizaje $\alpha_0(0, 1)$, radio de vecindad δ_0, épocas E_p y tamaño de la malla A.
Entrenamiento Serial:
for $e_p = 1 : E_p$ **do**
  for $t = 1 : T$ **do**
    1. Calcular las distancias del vector de entrada x_t con todos los vectores de pesos m_k con $k \in A$
    2. Encontrar la BMU
    3. Calcular el vecindario de la BMU
    4. Actualizar la BMU y sus vecinos usando la regla de actualización
  end for
  Ajustar el coeficiente de aprendizaje α y la vecindad δ
end for

Figura 2.10: Pseudocódigo del algoritmo del mapa auto-organizado con entrenamiento serial.

2.4.2. Medidas de calidad y precisión del mapa auto-organizado

Una de las características más importantes del mapa auto-organizado es la preservación de la topología del espacio de entrada por medio de sus relaciones de vecindad. Después de entrenar con los datos de entrada es necesario evaluar que tan bien se conservaron estas relaciones topológicas, así como la precisión del ajuste realizado por el modelo. Existen diferentes métricas para evaluar la calidad del entrenamiento, de las cuales las más utilizadas son el error de cuantización y el error topográfico.

El error de cuantización es una medida de la distancia promediada entre los vectores de entrenamiento y las neuronas BMU a las cuales fueron mapeados durante el proceso [24], si el valor es pequeño indica un buen ajuste de los datos de entrada. Ésta métrica es considerada como la medida de calidad básica de los mapas auto-organizados, no obstante es muy influenciada por el tamaño de la red o el número de épocas para el entrenamiento [24], además que la evaluación se realiza de forma local con las neuronas por ende no considera problemas de plegados incorrectos del mapa, en la ecuación 2.11 se muestra como se calcula.

$$Error_Q = \frac{1}{N}\sum_{i=1}^{N} ||x_i - m_c|| \tag{2.11}$$

El error topográfico es la medida de preservación de la topología, ya que describe que tan bien fue modelada la estructura de los datos de entrada. Esta métrica evalúa las discontinuidades locales en el mapa, ya que se calcula identificando a la mejor y la segunda mejor BMU de las entradas en el mapa evaluando su posición. En un mapa mal plegado y que se encuentre torcido de manera extraña, el error topográfico es grande incluso si el error de cuantización es pequeño. En la ecuación 2.12 se muestra como se calcula esta métrica de error.

$$Error_T = \frac{1}{N}\sum_{k=1}^{N} u(x_k), \quad donde \tag{2.12}$$

$$u(x_k) = \begin{cases} 1, & \text{si } BMU_1 \text{ y } BMU_2 \text{ son vecinos,} \\ 0, & \text{de otro modo,} \end{cases}$$

2.5. Técnicas de evaluación de grupos

El problema más importante cuando se trabaja con datos no etiquetados es conocer la estructura de la información y las relaciones entre los datos. Por lo tanto, generar grupos es una tarea fundamental cuando no se tiene información sobre la estructura que servirá para la clasificación de nueva información. Actualmente existen diversos algoritmos de aprendizaje no supervisado que agrupan las entradas utilizando diferentes enfoques: modelos basados en mezclas, minimización o maximización de distancias entre grupos, etc [19], sin embargo, las herramientas actuales no permiten conocer a detalle la cantidad de grupos que se pueden formar conservando las relaciones entre los datos. Por lo que existen diferentes tipos de agrupamiento que pueden ser clasificados en las siguientes categorías [25]:

- Agrupamiento jerárquico
- Agrupamiento por partición
- Agrupamiento basado en densidad
- Agrupamiento basado en modelos
- Agrupamiento basado en redes

El resultado de cada enfoque produce resultados diferentes aún con los mismos datos, por lo tanto, el objetivo de las técnicas de evaluación de la calidad del agrupamiento es encontrar la división de los datos que mejor ajuste al modelo [26]. El procedimiento de evaluar los resultados del agrupamiento en un algoritmo se conoce como validación por índices. Existen diferentes criterios de medición para evaluar y seleccionar el agrupamiento óptimo de la información tales como: compacidad y separación, es decir, grupos densos y bien separados.

Para la evaluación del resultado del agrupamiento existen tres diferentes criterios: externo, interno y relativo. Los criterios externo e interno están basados en métodos estadísticos por lo que la complejidad computacional se incrementa [26], en cambio el criterio relativo se fundamenta en la comparación de diferentes esquemas de agrupamiento eligiendo el mejor entre diferentes resultados.

2.5.1. Índice de Davies-Bouldin

Una de las técnicas más utilizadas para calificar el agrupamiento cuando no se conoce el *ground truth*, es decir, no se conoce la estructura de la información es el índice de Davies-Bouldin [27]. Este índice está basado en la medida de similitud de los grupos (R_{ij}) que se encuentran basados en la medida de dispersión de un grupo (s_i) y en la medida de disimilitud de agrupamiento (d_{ij}) [28]. Estas tres medidas se encuentran definidas en las ecuaciones 2.13 y 2.14.

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{2.13}$$

$$s_i = \frac{1}{||C_i||} \sum_{x \in C_i} d(x, v_i) \tag{2.14}$$

El índice de Davies-Bouldin mide el promedio de similitud entre cada grupo (C_i) y su más símil, entonces una medida baja significa un agrupamiento de grupos compactos y bien separados [28], en la ecuación 2.15 se define el índice de manera formal donde (n_c) es el número de grupos.

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i, \quad donde \tag{2.15}$$

$$R_i = \max_{j=1,...,n_c, i \neq j} (R_{ij}), i = 1, ..., n_c$$

2.5.2. Coeficiente de Silhouette

El método de Silhouette, es también otro de los métodos más utilizados cuando no se cuenta con datos etiquetados y se requiere medir la calidad de la agrupación, este proceso se lleva a cabo mediante la validación de la coherencia dentro de sus grupos [29]. Esta herramienta proporciona una representación gráfica y sucinta de que tan bien ha sido asignado cada dato de entrada con su respectivo grupo.

La evaluación se realiza por medio del valor de la silueta, la cual es una medida de similitud de un elemento y su propio grupo (cohesión) en comparación con otros cúmulos. La silueta tiene valores entre -1 a +1, donde un coeficiente cercano a 1 indica que el ejemplo de entrada esta bien emparejado con su grupo, es decir, esta agrupado correctamente. En cambio si la mayoría de los datos de entrada tienen un valor cercano a 0, indica que existen traslapes entre los grupos y si el valor es negativo entonces el ejemplo fue agrupado incorrectamente o la cantidad de grupos en la información puede ser mayor o menor. La distancia de las entradas a sus grupos puede ser calculada con cualquier métrica. El coeficiente de Silhouette se calcula usando la media de la distancia intra-grupal (a), es decir, la distancia entre el dato y todos los demás pertenecientes al mismo grupo y también por la distancia entre el dato y el grupo más cercano al cual no pertenece éste (b), en la ecuación 2.16 se define el calculo del coeficiente de manera formal [29].

$$S = \frac{b - a}{\text{máx}(a, b)} \tag{2.16}$$

2.5.3. Índice de representación múltiple

Uno de los más grandes retos cuando se intenta agrupar información sin etiquetar es elegir el número óptimo de grupos. Por lo que se han desarrollado diferentes formas de evaluar la calidad de los grupos basándose en los criterios de evaluación anteriormente mencionados. El índice de representación múltiple es un algoritmo independiente de la calidad de la partición generada, ya que está basado en el criterio relativo para la

validación de índices, es decir, grupos bien separados y compactos [26].

Sea un conjunto de datos S con formas arbitrarias, el índice de representación múltiple se ajusta bien a este tipo de grupos a través de la consideración de múltiples puntos representativos por grupo [26]. La multirepresentación está basada en el agrupamiento, ya que cada grupo puede ser representado por múltiples salidas. Este método reduce la complejidad computacional comparado con los métodos clásicos de agrupamiento jerárquico [25].

A continuación se define de manera formal el cálculo del índice de representación en donde un conjunto de datos esta dividido en c grupos. Sea $V_i = \{v_{i1}, v_{i2}, ..., v_{ir_i}\}$ un conjunto de puntos representativos que representan al i-ésimo grupo, donde r_i es el número de puntos representativos (datos) del i-ésimo grupo. Además sea $stdev(i)$ el vector de desviación estándar del i-ésimo grupo [25], entonces el p-ésimo componente de $stdev(i)$ ($stdev^p(i)$) se define en la ecuación 2.17.

$$stdev^p(i) = \sqrt{\sum_{k=1}^{n_i} (x_k^p - m_i^p)^2 / (n_i - 1)} \tag{2.17}$$

Donde n_i es el número de datos en el i-ésimo grupo, x_k es el dato perteneciente al i-ésimo grupo y m_i es la media del i-ésimo grupo. Entonces la desviación estándar promedio se define en 2.18.

$$stdev^p(i) = \sqrt{\sum_{i=1}^{c} ||stdev(i)||^2 / c} \tag{2.18}$$

Cuando la densidad intra-grupal es alta significa grupos bien separados [26], en la ecuación 2.19 se define.

$$Intra\_den(c) = \frac{1}{c} \sum_{i=1}^{c} \sum_{j=1}^{r_i} densidad(v_{ij}), \quad c > 1 \tag{2.19}$$

La $densidad(v_{ij})$ en 2.19 está definida por $\sum_{l=1}^{n_i} f(x_l, v_{ij})$ donde x_l pertenece al i-ésimo grupo, v_{ij} es el j-ésimo punto de representación en el grupo y $f(x_l, v_{ij})$ se define

en 2.20.

$$f(x_l, v_{ij}) = \begin{cases} 1, & ||x_l - v_{ij}|| \leq stdev, \\ 0, & \text{de otro modo} \end{cases} \tag{2.20}$$

La densidad inter-grupal es la que se encuentra en las zonas entre grupos [26], por lo que tiende a ser significativamente baja y se calcula como se muestra en 2.21.

$$Inter\_den(c) = \sum_{i=1}^{c} \sum_{j=1_{j\neq i}}^{c} \frac{||rep\_cercana(i) - rep\_cercana(j)||}{||stdev(i)|| + ||stdev(j)||}, \quad densidad(u_{ij}), \quad c > 1 \tag{2.21}$$

Donde $rep\_cercana(i)$ y $rep\_cercana(j)$ son el par de representaciones más cercanas del i-ésimo y j-ésimo grupo, mientras que u_{ij} es el punto medio entre estos puntos representativos como se muestra en la Figura 2.11.

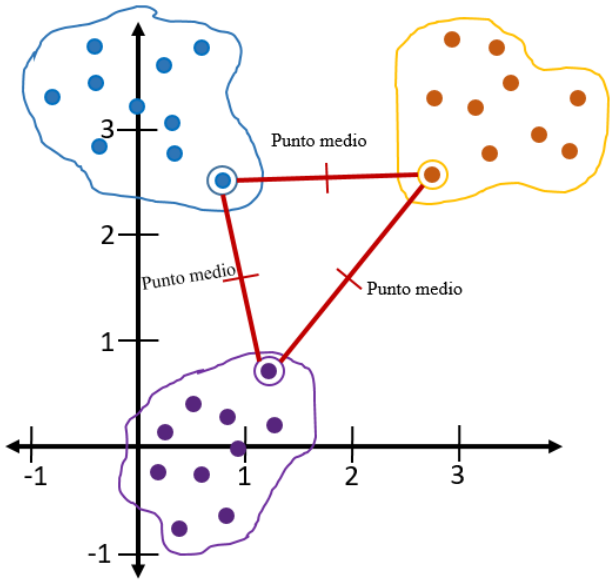


Figura 2.11: Puntos representativos más cercanos a los grupos vecinos.

La $densidad(u_{ij})$ está dada por $\sum_{k=1}^{n_i+n_j} f(x_k, u_{ij})$ donde x_k es el vector de entrada

perteneciente al i-ésimo y j-ésimo grupo y $f(x_k, u_{ij})$ se define en 2.22.

$$f(x_k, u_{ij}) = \begin{cases} 1, & ||x_k - u_{ij}|| \leq (||stdev(i)|| + ||stdev(j)||)/2, \\ 0, & \text{de otro modo} \end{cases} \tag{2.22}$$

Para evaluar la separación entre grupos se utilizaron ambas distancias inter-grupal e intra-grupal. Esta separación entre grupos se define formalmente en 2.23.

$$Sep(c) = \sum_{i=1}^{c} \sum_{j=1_{j \neq i}}^{c} \frac{||rep\_cercana(i) - rep\_cercana(j)||}{1 + Inter\_den(c)} \tag{2.23}$$

El índice de representación múltiple alcanza su máximo cuando c es el óptimo [26] y se calcula como se muestra en la ecuación 2.24.

$$IRM = Intra\_den(c) * Sep(c) \tag{2.24}$$

2.6. Métricas de evaluación para la clasificación de datos

En los problemas de clasificación de datos con aprendizaje de maquina, se requieren métricas para medir el desempeño del modelo desarrollado con los datos que se desean clasificar y observar que tan bien el modelo aprendió la estructura. Por lo tanto, se describen algunas medidas de evaluación utilizadas para evaluar la metodología desarrollada en el trabajo de investigación:

- Exactitud
- Precisión
- Sensibilidad
- Especificidad
- Valor F1

La exactitud es la métrica definida como el total de elementos clasificados correctamente. Esta relación se muestra en la ecuación 2.25 donde :

- Verdadero positivo (VP)
- Verdadero negativo (VN)
- Falso positivo (FP)
- Falso negativo (FN)

Esta métrica mide la calidad entre los clasificadores, en un valor entre 0 y 1. Además, es natural e intuitiva y se utiliza ampliamente para medir el desempeño de los modelos. Sin embargo, solamente trabaja bien para clases balanceadas, ya que al no estar balanceadas el que tiene más oportunidad de ser predicho será de mayor cantidad de ejemplos en la base de datos [30].

$$Exactitud = \frac{VP \ + \ VN}{VP \ + \ FP \ + \ VN \ + \ FN} \tag{2.25}$$

La precisión se define como número de elementos identificados correctamente como positivos de un total de elementos identificados como positivos. Esta relación se muestra en la ecuación 2.26

$$Precisión = \frac{VP}{VP \ + \ FP} \tag{2.26}$$

La sensibilidad representa al número de elementos identificados correctamente como positivos del total de verdaderos positivos [31]. En la ecuación 2.27 se muestra el cálculo de esta métrica.

$$Sensibilidad = \frac{VP}{VP \ + \ FN} \tag{2.27}$$

La especificidad es el número de vectores correctamente identificados como negativos con respecto al total de negativos [31]. En la ecuación 2.28 se muestra el cálculo de esta métrica.

$$Especificidad = \frac{VN}{VN \ + \ FP} \tag{2.28}$$

El valor F1 es la media harmónica entre la precisión y la sensibilidad la cual está comprendida entre 0 y 1. Este valor mide que tan preciso es el clasificador, así como que tan robusto es evitando perder un número significativo de instancias, encontrando un balance entre precisión y sensibilidad [31]. En la ecuación 2.29 se muestra el cálculo de esta métrica.

$$F1 = 2 * \frac{1}{\frac{1}{precisión} + \frac{1}{sensibilidad}} \tag{2.29}$$

Capítulo 3

Agrupamiento de datos con AGs y mapas auto-organizados

La metodología del trabajo de investigación ha sido dividida en cinco secciones principales, en cada apartado se abordan los métodos y procesos desarrollados para implementar los modelos de agrupamiento y clasificación:

- Metodología 1: agrupamiento por algoritmos genéticos
- Metodología 2: agrupamiento por mapas auto-organizados
- Herramientas de funcionalidad y visualización de resultados
- Pre-procesamiento de la base de datos sísmicos
- Evaluación de la metodología de clasificación

La siguiente metodología se desarrolló para clasificar datos empleando técnicas de aprendizaje de máquina. Se implementaron dos formas diferentes para resolver el problema de la creación de los grupos para la clasificación (etiquetas), además se trabajó con bases de datos etiquetadas y no etiquetas. Antes de iniciar el proceso de generar las etiquetas, las bases de datos fueron pre-procesadas, en donde se normalizaron y estandarizaron los datos. Después, con las redes neuronales y los algoritmos genéticos se crearon los grupos por medio del aprendizaje no supervisado.

Seguidamente los datos no utilizados en el entrenamiento se emplearon como conjunto de validación del modelo. Por otra parte, se utilizaron bases de datos de prueba para medir el desempeño del algoritmo utilizando diferentes métricas de evaluación.

3.1. Metodología 1: agrupamiento por algoritmos genéticos

Actualmente existen diferentes técnicas para agrupar información en donde se tiene como objetivo optimizar criterios de distancia o maximizar-minimizar funciones. Puesto que existen procesos de optimización dentro del flujo de trabajo de los algoritmos, en ocasiones pueden converger a mínimos o máximos locales que no representan a la solución óptima para resolver el problema por la forma de inicialización de sus valores [32].

Partiendo de esta idea se utilizó al algoritmo genético como herramienta de aprendizaje no supervisada para etiquetar información de las bases de datos. Primero, se definió la codificación del problema dentro de los individuos (cromosomas) de la población, en la Figura 3.1 se observa esta codificación en donde cada vector de entrenamiento está representado por su posición en el cromosoma de izquierda a derecha en orden ascendente marcados como S_N, mientras que en los genes del cromosoma se indica el grupo al cual pertenece cada uno. El número de grupos que se pueden formar durante el entrenamiento es un parámetro libre que el usuario configura basándose en el análisis de evaluación por índices.

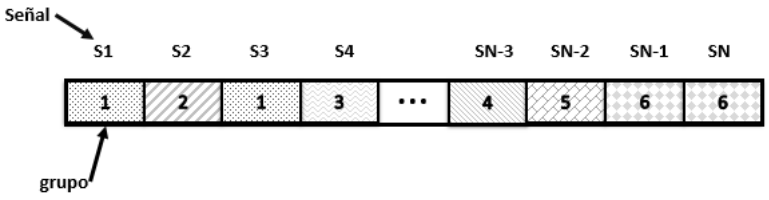


Figura 3.1: Codificación del problema en los individuos de la población.

Las etiquetas asignadas a cada vector al principio del proceso evolutivo fueron seleccionadas de forma aleatoria. Por otro lado, se planteó la función objetivo con base al problema de agrupamiento como se muestra en la ecuación 3.1, en donde $D_{C1}, D_{C2}, .., D_{Cn}$ representan las sumas de las distancias entre todos los elementos X (vectores de entrenamiento) que pertenecen a cada grupo como se observa en la ecuación 3.2.

$$f_{obj}min(\frac{D_{C1} + D_{C2} + ... + D_{Cn}}{n}) = \text{mín } \frac{\sum_{i=1}^{n} D_{Ci}}{n} \tag{3.1}$$

$$D_{Ci} = \sum_{j=1}^{k} \sum_{r=1}^{p} \|X_j - X_r\| \tag{3.2}$$

La función objetivo 3.1 minimiza el promedio de las distancias entre todos los vectores pertenecientes a cada grupo. Para los individuos que obtuvieron la mejor y peor evaluación del vector de aptitud de la población (resultados de la función) se les aplicó el principio de elitismo como se mencionó en el Capítulo 2, por lo que al menos dos copias del mejor individuo se conservan en la siguiente generación.

Por otra parte, durante el ciclo evolutivo se generan patrones que permiten caracterizar al grupo después del ciclo evolutivo, emulando la etapa de actualización utilizada por el algoritmo K-medias [33], éstos centroides se actualizan en cada generación para obtener el vector de características que será utilizado para la clasificación de datos nuevos. En la ecuación 3.3 se muestra el cálculo de los centros de los grupos en donde los X_j son los elementos que pertenecen a cada grupo C_i formado durante el proceso.

$$C_i = \frac{\sum_{j=1}^{k} X_j}{k} \tag{3.3}$$

El método de selección utilizado fue el de torneo, utilizando dos fases de eliminación ya que se observó que mejora la velocidad de convergencia hacia la solución. Se utilizó la codificación entera, pues la codificación de los grupos son números enteros, es decir, en cada gen se indicó el número del grupo al cual pertenecía el vector de entrenamiento asociado. La operación de cruza se realizó con el operador de cruce multi-punto y para la operación de mutación se utilizó el método por intercambio de gen, ambos descritos

en el Capítulo 2. Con el objetivo de agrupar a los vectores de entrenamiento y crear las clases (*ground truth*) que son descritas a partir de los patrones calculados (centroides), se definieron de forma fija los parámetros de operación del algoritmo genético para estandarizar los procesos de entrenamiento, éstos se muestran en la Tabla 3.1.

Tabla 3.1: Parámetros de configuración predefinidos para entrenar con el algoritmo genético.

Parámetro	Configuración
Tamaño de la población	1000
Longitud de individuo	Codificación del vector
Número de generaciones	10000
Probabilidad de cruza	90 %
Probabilidad de mutación	10 %

3.2. Metodología 2: agrupamiento por mapas auto-organizados

Se implementó una metodología basada en redes neuronales de tipo mapas auto-organizados, para definir e identificar grupos en los mapas de neuronas de la SOM. Estos grupos se pueden etiquetar después del entrenamiento de la red y el agrupamiento de las neuronas con sus respectivos grupos, para proceder con la clasificación de datos que no han sido utilizados en el entrenamiento. Antes de describir la metodología para la creación de clases en el mapa de neuronas, es importante explicar las etapas previas al proceso de entrenamiento de la red neuronal.

En primer lugar, los datos fueron transformados y convertidos en vectores de entrenamiento para las entradas de la red, como se describió en el capítulo 2 en el apartado de pre-procesamiento. Antes de la etapa de entrenamiento de los mapas auto-organizados se configuraron los parámetros de operación del mapa para entrenar. En la configuración se fijaron algunas configuraciones para comparar el desempeño de los modelos con parámetros estándar para todos, en la Tabla 3.2 se muestra la configuración estándar.

Tabla 3.2: Parámetros de configuración para entrenar con los mapas auto-organizados.

Parámetro	Configuración
Tamaño del mapa	Variable
Épocas	1000
Tasa de aprendizaje	0.01, 0.1 y 0.2
Radio de vecindad	(Tamaño del mapa) x 0.1
Topología	Hexagonal

Después de configurar los mapas se realizaron los entrenamientos evaluando su desempeño mediante el error de cuantización y el error topográfico. Medir el desempeño de una SOM no es fácil, es por eso que existen diferentes métricas para hacerlo y observar que tan bueno fue el aprendizaje de la topología desde enfoques diferentes, sin embargo, se eligió el error topográfico para decidir qué mapa iba a ser seleccionado después de los diez entrenamientos para medir el desempeño del clasificador.

Como se mencionó en el Capítulo 2 cada neurona de la red está conformada por coeficientes con la misma dimensión que los datos de entrenamiento. Estas neuronas representan de forma individual un patrón característico aprendido durante el entrenamiento. Haciendo uso de esta característica de los mapas auto-organizados, una vez entrenada la red se dividió por regiones para formar grupos representados por neuronas con coeficientes con características similares, el número de grupos que se forman es seleccionado por el usuario previamente. Si no se conoce el cantidad de clases en las que se dividirá el mapa, se realiza una selección manual del número por medio de los índices de evaluación de calidad en el agrupamiento como se explicó en el Capítulo 2.

3.2.1. Creación de clases etapa 1 de 3: creación de grupos

Después de concluir con el proceso de entrenamiento del mapa, para la primera etapa se calculó una matriz de distancias euclidiana diagonal superior entre los coeficientes

de las n neuronas, como se muestra en la ecuación 3.4.

$$DN = \begin{pmatrix} DN_{1,1} = 0 & DN_{1,2} & \cdots & DN_{1,n} \\ 0 & DN_{2,2} = 0 & \cdots & DN_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & DN_{n,n} = 0 \end{pmatrix} \tag{3.4}$$

La matriz de distancias representa una medida de similitud entre neuronas basado en la diferencia entre sus coeficientes. Después de la creación de la matriz el usuario necesita configurar dos parámetros: número de clases y distancia máxima entre neuronas. El primer parámetro determina la cantidad de grupos formados al final de las tres etapas, mientras que el segundo establece la distancia máxima entre neuronas para crear los primeros grupos durante esta etapa, por medio de dos ajustes denominados ajuste "grueso" y "fino". Después de configurar los parámetros, el proceso inicia con la primera neurona $N_{1,1}$ que se denomina como la neurona raíz. A partir de ella todas las neuronas que cumplan que su distancia sea menor a la distancia máxima configurada por el usuario formarán parte del primer grupo. Este proceso de expansión del grupo se detiene al no encontrar más neuronas que cumplan con la condición.

Posteriormente, el proceso continua creando el segundo grupo a partir de la primera neurona vecina superior sin agrupar, que se convierte en la siguiente neurona raíz. Al finalizar la creación del nuevo grupo se realiza un proceso de verificación entre el grupo creado y el anterior, utilizando a las neuronas raíces de ambos como centroides para medir la distancia entre todas la neuronas pertenecientes a los dos grupos y sus centros, reagrupando a las neuronas que cumplen que están más cercanas al centro del grupo vecino que al que pertenecen, en la Figura 3.2 se muestra el proceso de creación de grupos.

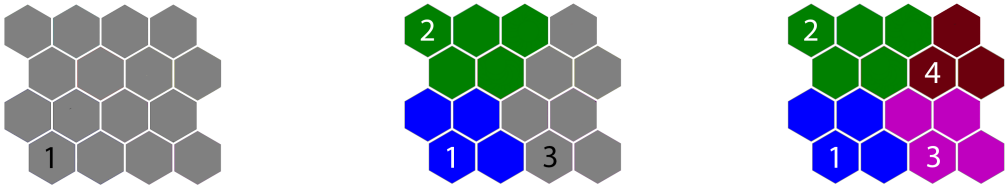


Figura 3.2: Creación de grupos en el mapa de neuronas durante la primera etapa.

Este proceso es iterativo, generando los grupos que cumplen con la condición de distancia entre las neuronas raíces y las que no se encuentran agrupadas, el número máximo de grupos que se pueden formar en esta primera etapa es igual al número de neuronas en el mapa. Después de dividir el mapa en todos los grupos que se pudieron formar, los centroides de cada grupo se reasignan a otra neurona. Para esto se promedian los pesos de todas las neuronas pertenecientes al mismo grupo y se selecciona a la neurona más cercana al promedio como nuevo centroide. Terminado el proceso de reasignar los nuevos centros, se realiza una última inspección para confirmar que las neuronas están correctamente agrupadas como se describió en el proceso intermedio de creación de grupos nuevos. El proceso de verificación se asemeja a K-medias durante la creación de centroides a partir de los datos y el chequeo de la pertenencia de estos a los grupos correctos, sin embargo, la diferencia entre los métodos radica en la asignación del centro a la neurona nueva, en la Figura 3.3 se muestra esta última verificación.

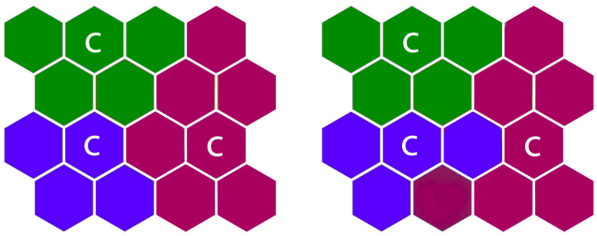


Figura 3.3: Verificación final de los grupos formados en el mapa de neuronas.

Durante los procesos de verificación entre la creación de grupos y al final de este proceso se observó que la mayoría de las neuronas reagrupadas se encuentran en las

fronteras de los grupos grupos, ya que existen valores muy similares de distancia y por eso al principio fueron asignadas a otros grupos, por lo que es necesaria esta última verificación para delimitar mejor las fronteras.

3.2.2. Creación de clases etapa 2 de 3: Unión y creación de proto-grupos

Con los grupos ya formados en el mapa de neuronas se procedió a unirlos utilizando a los siguientes descriptores estadísticos que caracterizan cada grupo de forma individual:

- Promedio
- Desviación estándar
- Moda
- Rango

Para ello se calcularon los estadísticos para cada grupo y se generó una matriz de distancias entre ellos, de forma similar a la calculada en la primera etapa como se muestra en la ecuación 3.5, pero midiendo la similitud entre grupos como se observa en la ecuación 3.6, en donde $DE_{i,j}$ representa la distancia entre los estadísticos de las neuronas i j, e_1 es el promedio, e_2 es la desviación estándar, e_3 es la moda y e_4 es el rango.

$$DE = \begin{pmatrix} DE_{1,1} = 0 & DE_{1,2} & \cdots & DE_{1,n} \\ 0 & DE_{2,2} = 0 & \cdots & DE_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & DE_{n,n} = 0 \end{pmatrix} \tag{3.5}$$

$$DE_{i,j} = |e_{1,i} - e_{1,j}| + |e_{2,i} - e_{2,j}| + |e_{3,i} - e_{3,j}| + |e_{4,i} - e_{4,j}| \tag{3.6}$$

Antes de iniciar el proceso de unión entre grupos el usuario tiene que configurar otro parámetro denominado porcentaje de similitud basado en la matriz de distancias de estadísticos.

Debido a que la unión entre grupos podría generarse entre diferentes grupos, se limitó a una unión por cada grupo para evitar que al final de esta etapa queden menos grupos que el número seleccionado por el usuario en la etapa previa. En caso de haber más de un grupo que cumple con la característica, solamente se dará la unión con el grupo de mayor similitud y que no se encuentre unido previamente. Esta medida tiene la finalidad de controlar la cantidad de uniones de grupos al finalizar el proceso, creando así prototipos previos antes de la etapa final. En la Figura 3.4 se muestra el proceso de unión de grupos, el grupo marcado con R es el grupo que se unirá a cualquiera de los tres grupos aledaños en los cuales se muestra el porcentaje de similitud con respecto a este, uniéndose al más símil.

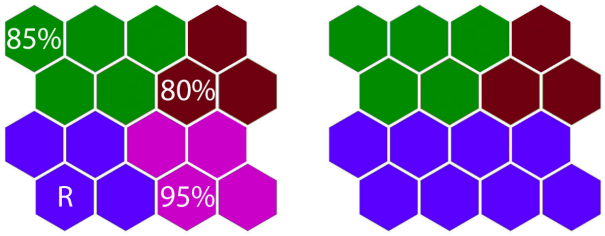


Figura 3.4: Unión y creación de proto-grupos de neuronas basado en los estadísticos y el porcentaje de similitud.

3.2.3. Creación de clases etapa 3 de 3: Unión de grupos

Para la tercera y última etapa, primero se verifica cuantos agrupamientos fueron creados en la etapa anterior, en caso de ser igual al número definido por el usuario al inicio de las etapas este proceso finaliza. De otro modo, si el número de grupos es mayor al configurado, el flujo de trabajo continúa seleccionando de los grupos formados en la segunda etapa a los que mayor activación de neuronas obtuvieron durante el entrenamiento, denominando a estos como grupos bases. El mapa de activación es un parámetro que se obtiene del entrenamiento, en la Figura 3.5 se observa el número de activaciones para cada neurona las cuales fueron codificadas utilizando la técnica de normalización del mínimo-máximo codificada en una escala del 0 al 1.

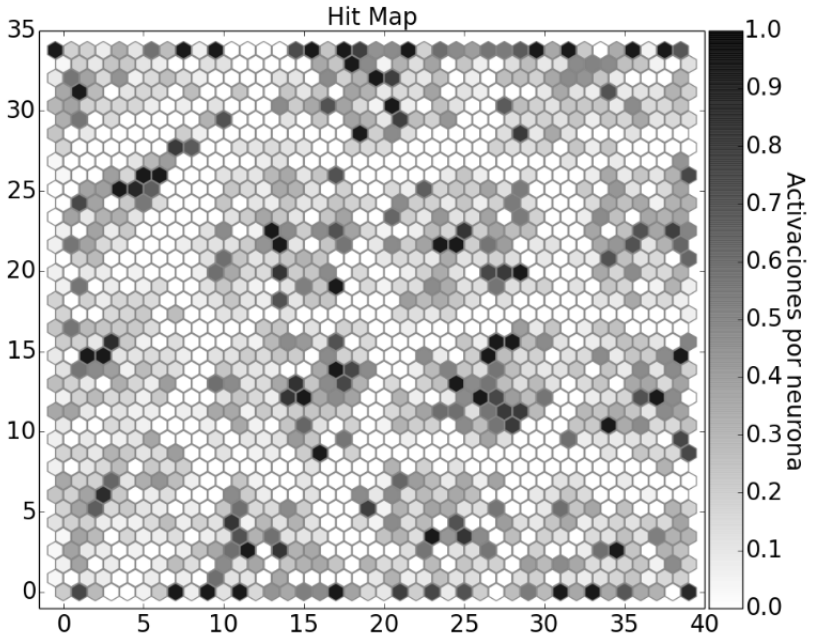


Figura 3.5: Mapa de activación de neuronas después del proceso de entrenamiento.

Los grupos libres se irán uniéndose a los bases durante el último proceso de unión. Esta separación se realiza ya que los grupos con más activaciones concentran características relevantes que ayudan a la diferenciación entre ellos, es decir, concentran los patrones que mejor los caracterizan, así también se observó que los que fueron activados en menor cantidad representan las zonas límites entre grupos o concentran zonas con patrones muy similares que no presentaban variaciones significativas entre los patrones.

Después de identificar a los grupos bases y libres se calculó el índice de representación múltiple entre cada grupo libre y todos los base, el cual está explicado en el Capítulo 2. Esto evalúa la unión entre grupos en donde valores de índices bajos indican mala separación entre ellos y alta aglomeración de datos, lo que significa que el grupo original fue dividido durante el proceso de creación. Por lo tanto, la unión se realiza entre los grupos que obtuvieron el menor índice de representación múltiple, generando así un nuevo grupo en el mapa. En ésta etapa el proceso termina cuando el número de grupos final es igual al seleccionado por el usuario.

3.3. Herramientas de funcionalidad y visualización de resultados

Se utilizó la biblioteca de mapas auto-organizados denominada MiniSom [34] desarrollada por Giussepe por su facilidad de uso, eficiencia en el proceso de entrenamiento y versatilidad al configurar el tamaño del mapa, el radio de vecindad, la función de vecindad, la configuración de la topología de la red y la tasa de aprendizaje. Además, el tipo de entrenamiento implementado en la biblioteca es por lotes (*batch*), lo que incrementa la velocidad del proceso pero requiere más épocas para lograr un mejor resultado al entrenar.

La biblioteca MiniSom no tiene implementada la métrica de error topográfico para mapas con topología hexagonal, así que como parte de este trabajo de investigación se implementó el algoritmo para calcular dicho error y evaluar así el entrenamiento de los mapas. Como se mencionó en el capítulo 2, el error topográfico mide la preservación de la topología de los datos utilizando la posición de la primera y la segunda neurona ganadora como referencia, por lo tanto se localizó la posición de la segunda neurona ganadora mediante los índices de posición de las neuronas en el mapa. Se implementó una función de verificación como se muestra en la Figura 3.6, en donde para columnas pares e impares con respecto a los índices existen diferentes formas de calcular el índice de la neurona vecina en una topología hexagonal.

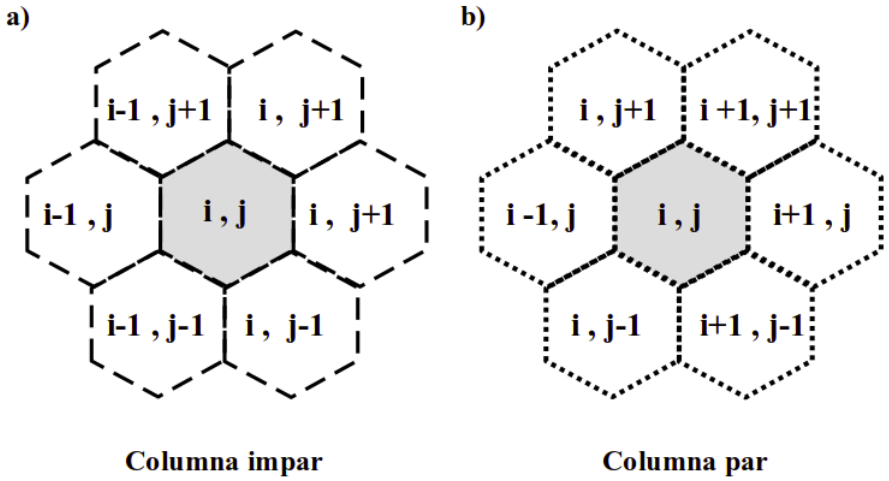


Figura 3.6: a) Índices de las neuronas vecinas en una columna impar b) Índices de las neuronas vecinas en una columna par.

Después del proceso de entrenamiento del mapa auto-organizado se crea un mapeo N-dimensional a uno bidimensional representado en unidades llamadas neuronas [35]. Este mapeo conserva las relaciones topológicas de espacios N-dimensionales en las neuronas que se encuentran ordenadas en una malla. En una topología hexagonal cada una de las neuronas se encuentra rodeada de seis neuronas vecinas con pesos N-dimensionales en donde visualizar estas relaciones espaciales dentro de la malla no es una tarea trivial. Ultsch [35] desarrolló una metodología de visualización de datos multidimensionales, en donde unifican estas distancias en una medida que puede ser representada con un valor en una matriz de una capa unitaria, conocida como matriz de distancia unificada [35], ésta es utilizada para desplegar las similitudes/disimilitudes de las neuronas y sus vecinas, en la Figura 3.7 se observa la relación de vecindad entre neuronas.

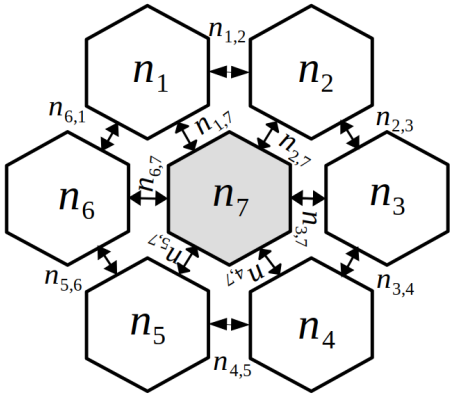


Figura 3.7: Vecindad de las neuronas en una topología hexagonal.

La matriz resultante muestra las disimilitudes entre los datos de entrada, así como cambios grandes de distancia que si son desplegados de forma tridimensional formarían montañas entre los datos de entrada y en forma bidimensional se interpreta como cambios de color entre los diferentes conjuntos. Esta característica de la técnica de visualización de mapas auto-organizados permite la detección en algunos casos de grupos en los datos, en la Figura 3.8 se observa una matriz de distancia unificada en dos dimensiones obtenida después de entrenar con una base de datos en donde la diferencia de distancias se encuentra codificada en una barra para una visualización más intuitiva.

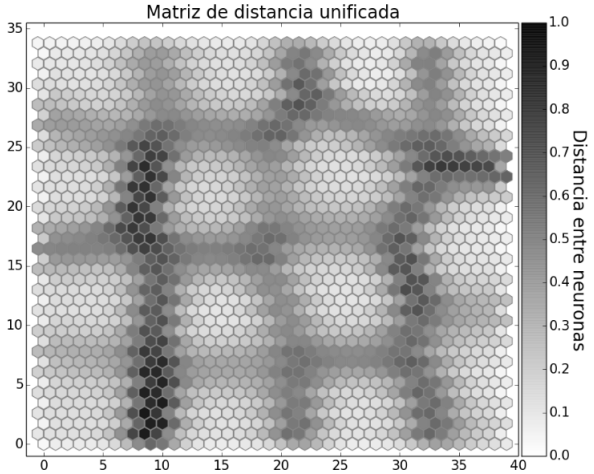


Figura 3.8: U-matriz con grupos separados por zonas de mayor distancia cercanas a 1 y a 0 si la separación es menor.

3.4. Pre-procesamiento de la base de datos sísmicos

Antes del entrenamiento del modelo es necesario pre-procesar la información como se mencionó en el capítulo 2, esta etapa previa es utilizada para darles formato, manipular los datos faltantes, transformar la información en vectores de entrada o para normalizar los datos.

La base de datos sísmicos proviene de un análisis realizado en una región donde se quiere identificar y clasificar zonas similares de interés dentro de un cubo sísmico. Los datos que se usaron para la clasificación corresponden a 9 secciones verticales de un cubo sísmico, cada uno de estos planos consta de 460 señales sísmicas de 625 muestras. Estas secciones fueron previamente pre-procesadas para formar lo que se conoce como datos sísmicos de trazas apiladas, esto es, cada una representa el promedio de varias trazas de una misma región, con lo que se reduce el ruido [36], cada señal es el promedio de 40 trazas vecinas, encapsulando la información de las zonas aledañas y organizándola en matrices de 460 x 625 para formar imágenes sísmicas verticales del subsuelo, como se muestra en la Figura 3.9, donde el nivel de gris indica la intensidad o energía de la señal sísmica.

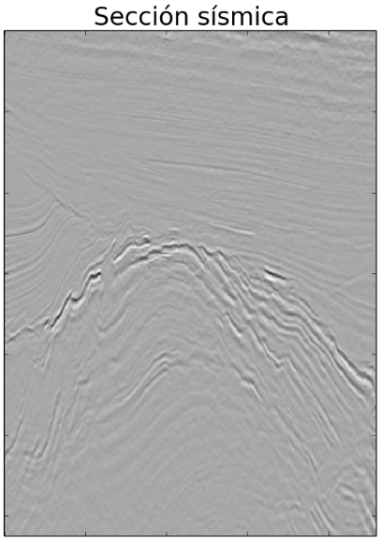


Figura 3.9: Sección sísmica para clasificar.

La base de datos sísmicos no está etiquetada, por lo que se utilizaron las redes neuronales y los algoritmos genéticos para generar el *ground truth* utilizando un subconjunto de los datos. Para el grupo de entrenamiento, primero se seleccionó de cada plano sísmico una franja central de datos equivalente a un 20 % de la información contenida en cada una de las nueve secciones sísmicas como se muestra en la Figura 3.10.

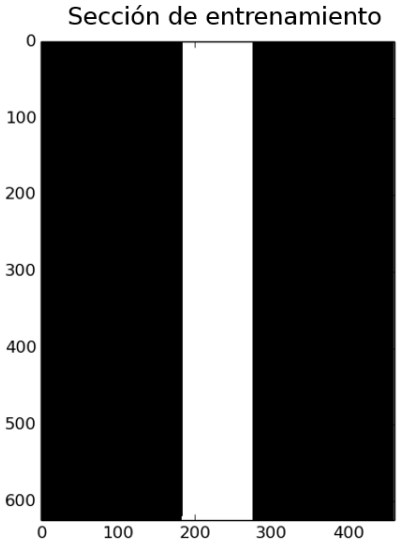


Figura 3.10: Franja central extraída de las secciones sísmicas para agrupar los datos y crear las clases.

Por otra parte, se observó que las amplitudes entre secciones variaban mucho por lo que se normalizó la información usando la técnica del máximo global, quedando en un rango entre 1 a -1 los datos sísmicos. La estructura de la base de datos de entrenamiento fue organizada de tal forma que cada fila representa a un vector de entrenamiento, los cuales son obtenidos mediante una transformación de los datos en las secciones sísmicas.

La primera opción utilizada para la transformación fue un vector deslizante de 25 muestras por traza con un traslape del 20 % al 80 %, sin embargo, debido a que el propósito es identificar regiones en el subsuelo y este es muy heterogéneo, es decir, hay muchas zonas con características físicas similares, se optó por utilizar una ventana deslizante de 5 x 5 para conservar estas relaciones entre regiones adyacentes. Al utilizar esta técnica por ventanas las relaciones estructurales del subsuelo se conservan

dándole sentido al análisis de loas datos sísmicos. Además, es importante señalar que la combinación del tamaño de la ventana y el nivel de traslape entre ventanas, afecta directamente en la resolución del análisis, en el ancho de banda de las señales que se van a analizar y en el volumen de datos que se van a procesar. Ventanas grandes con poco traslape conducen a bases de datos más pequeñas, pero se pierde resolución espacial. Por otra parte, ventanas pequeñas permiten detectar características más especificas, pero el volumen de datos crece de manera importante.

Primero, se hicieron análisis variando el tamaño de ventana, donde se optó por tamaños pequeños que permitan observar estas estructuras físicas de forma nítida en la imagen, por lo que se eligieron dos tamaños estándares: 10 x 10 y 5 x 5 píxeles. Para conservar los patrones característicos de una región homogénea se requiere que ese patrón sea continuo y no sea cortado al momento de segmentar las secciones y generar los vectores de entrenamiento, por ende se propusieron dos porcentajes de traslape entre ventanas para conservar estas características: 60 % y 80 %. Es correcto pensar que un traslape mayor genera información redundante que se ingresa al entrenamiento y se traduce en costo computacional, sin embargo, al contar con información duplicada la red aprende con más detalle esos patrones y las relaciones entre zonas adyacentes, evitando perder las patrones interesantes en la señal.

Estos dos parámetros configurables para la transformación de los datos fueron probados y evaluados por medio de dos métricas: error de cuantización y error topográfico. Se observó que los mejores resultados se presentaban con un traslape de 80 % y una ventana de 5 x 5 debido a que ambas métricas de error eran las más pequeñas en mapas entrenados con este pre-procesamiento, además la resolución de las imágenes obtenidas después del agrupamiento y clasificación permitían observar de forma nítida las regiones similares y existía congruencia en las estructuras geológicas al comparar con las imágenes sin clasificar (grises). Los resultados se codificaron en una secuencia de colores que representa la pertenencia a una clase.

En la Figura 3.11 se observa la segmentación de las imágenes sísmicas en ventanas de 5 x 5 píxeles, en donde cada ventana viene codificada en colores diferentes y los

píxeles están marcados con S1, S2, S3 y S4 para visualizar a que ventana pertenecen y donde se presentan los traslapes entre ventanas. El proceso de segmentación se realizó de forma iterativa de arriba hacia abajo y de izquierda a derecha, en la Figura 3.11 se muestra solamente la segmentación vertical de las ventanas.

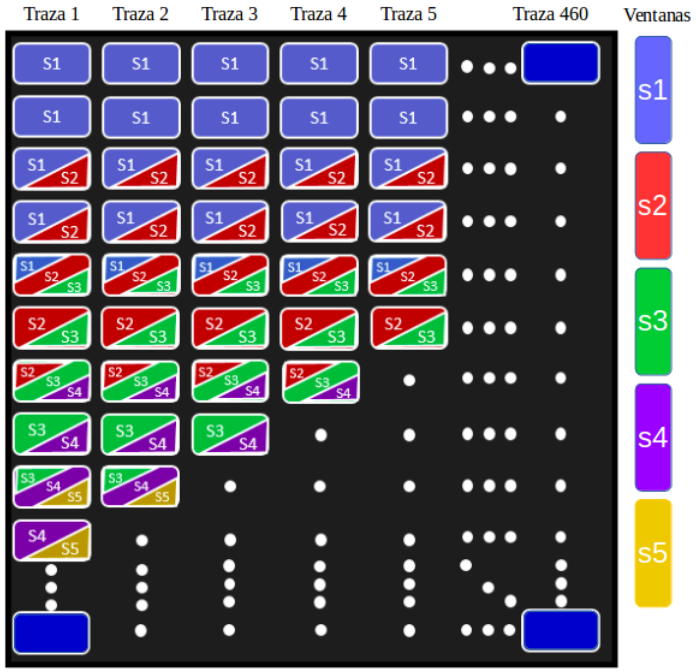


Figura 3.11: Proceso de transformación de los datos para construir la base de datos de entrenamiento.

Ahora bien si la información fue segmentada en ventanas de 25 muestras cada una, fue necesario convertirlas en vectores de entrada para el agrupamiento. Por lo tanto, las ventanas se pre-procesaron dividiéndolas por filas y concatenándolas en un solo vector de datos como se muestra en la Figura 3.12, en donde los números en cada píxel ayudan visualmente a identificar el orden espacial de cada uno en los vectores de entrenamiento.

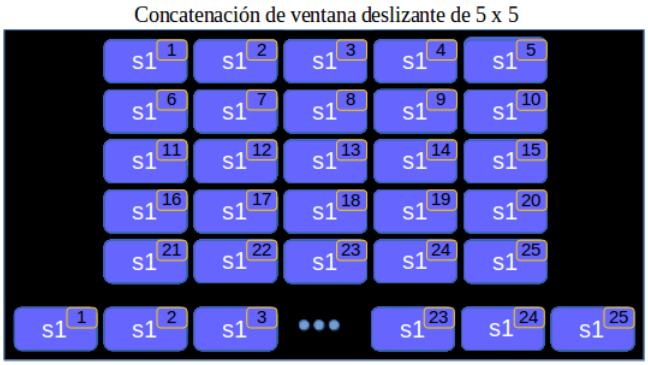


Figura 3.12: Proceso de transformación de la ventana al vector de entrenamiento.

La información muestreada equivale a las amplitudes de las señales sísmicas obtenidas con los geófonos, entonces se realizó un análisis en amplitud y a partir de ellos también se analizaron en potencia, en la ecuación 3.7 se muestra la transformación en potencia de los datos X.

$$X_{pot} = X^2 \tag{3.7}$$

Las amplitudes de onda capturadas son de magnitud positiva y negativa y representan a los ciclos de cada señal capturada, es por eso que analizar las señales en amplitud sin pre-procesamiento permite observar con más detalle las transiciones entre regiones símiles, que ocurre durante el cambio en el signo de cada traza. Por otro lado, si se desea observar y remarcar zonas con grandes cambios en amplitud de la señal (energía) es necesario realizar un análisis en potencia, ya que las zonas muy similares son homogeneizadas difuminando en la imagen esos detalles.

3.5. Evaluación de la metodología de clasificación

Para evaluar la metodología de clasificación por etapas con la red neuronal se dividieron las pruebas en dos secciones: pruebas con bases de datos etiquetadas y pruebas con bases de datos no etiquetadas. Las primeras evalúan la calidad del modelo con distintas métricas. En cuanto a las pruebas con la base de datos no etiquetada (da-

tos sísmicos), primero se entrenó el modelo con un porcentaje del 20 % de los datos, posteriormente por medio de un análisis con los índices de calidad del agrupamiento se eligió el número de clases que se formarían con la metodología. Por último, se clasificó la información que no había sido utilizada para crear las clases. En cuanto a los entrenamientos realizados con la red neuronal se configuraron los parámetros como se muestra en la Tabla 3.2.

Las pruebas de evaluación que se realizaron fueron basadas en los métodos de Monte Carlo para seleccionar a los conjuntos de validación y entrenamiento empleando procesos completamente aleatorios, además se realizó la validación cruzada "K-Fold" para medir el desempeño del modelo. Las pruebas se realizaron en un servidor Intel i7 de 9na generación con 8 núcleos y 64 GB de RAM, que cuenta también con una tarjeta gráfica Nvidia GeForce RTX 2080 Ti. A continuación, se listan y describen los procedimientos realizados para evaluar las bases de datos etiquetadas así como la composición de éstas, además se explica la metodología para clasificar la base de datos sísmicos.

3.5.1. Iris

Esta base de datos es una de las más conocidas y utilizadas en la literatura de reconocimiento de patrones [37], ya que presenta características sencillas para realizar una prueba de clasificación. La base de datos contiene tres clases de plantas iris: Setosa, Versicolour y Virginica, para cada clase se cuentan con 50 ejemplos de cuatro atributos que se listan a continuación:

- Longitud del sépalo en cm
- Ancho del sépalo en cm
- Longitud del pétalo en cm
- Ancho del pétalo en cm

Una de las características más interesante de esta base de datos es que una de las clases es linealmente separable de las otras dos, pero las otras no pueden ser separadas

una de otra de forma lineal. Para evaluar el modelo de clasificación se entrenó la red neuronal con una mapa de tamaño de 4 x 4 neuronas normalizando los datos con la técnica de mínimo-máximo, así también se entrenó con los datos sin normalización para comparar el desempeño. Esta base de datos fue utilizada para evaluar el modelo por primera vez por sus características descritas.

3.5.2. Vinos

La base de datos de vinos fue creada después de analizar químicamente tres diferentes cultivos de vinos crecidos en la misma región en Italia [38]. Este análisis proporciona cantidades de 13 elementos diferentes encontrados en los tres cultivos analizados. Por lo tanto, los datos tienen trece atributos que los caracterizan y son utilizados para tareas de clasificación, entre los elementos que constituyen a las clases se encuentran: alcohol, ácido málico, ceniza, magnesio, flavonoides, entre otros.

Esta base de datos es ampliamente utilizada para probar nuevos algoritmos de clasificación, ya que el problema de clasificación está bien definido y además la extensión de la base de datos es pequeña conformada por 178 datos de trece dimensiones cada uno y es un problema de clasificación multi-clase (tres cultivos de vino). Para el entrenamiento se utilizó un mapa de tamaño de 4 x 4 neuronas con topología hexagonal. Si bien es una base compacta y es muy utilizada para pruebas de clasificación, la complejidad aumenta con la dimensionalidad, además de que las clases no se encuentran balanceadas. Por otra parte, los datos presentan relaciones lineales fuertes entre sí y no están estandarizadas, por lo que se realizó una estandarización de estos. En las Figuras 3.13 y 3.14 se observa una proyección bidimensional con tSNE de la base de datos de vinos antes y después de estandarizar los datos. Se observa que después de la estandarización los datos se separan mejor, facilitando el proceso de clasificación.

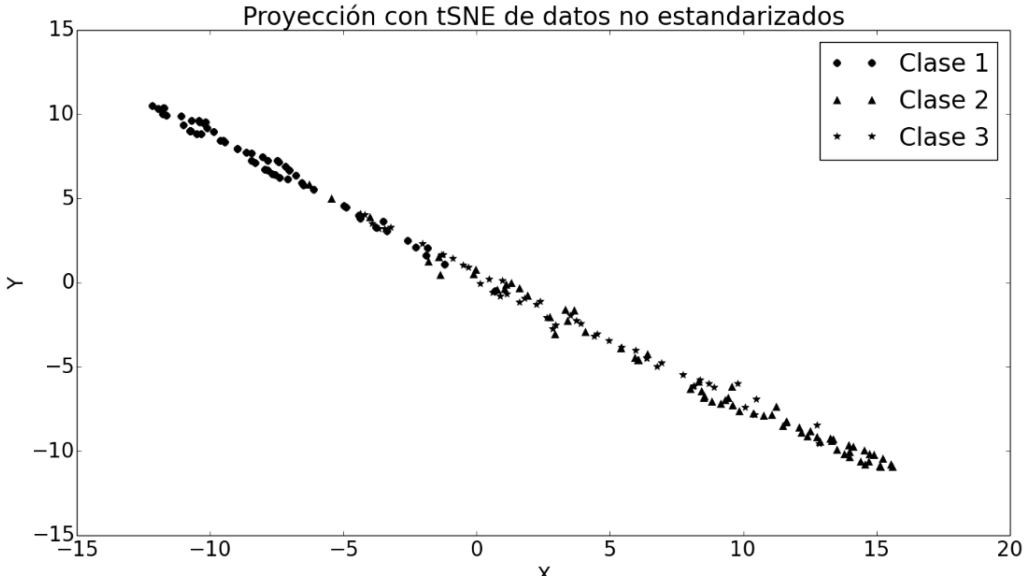


Figura 3.13: Proyección de datos no estandarizados de la base de vinos.

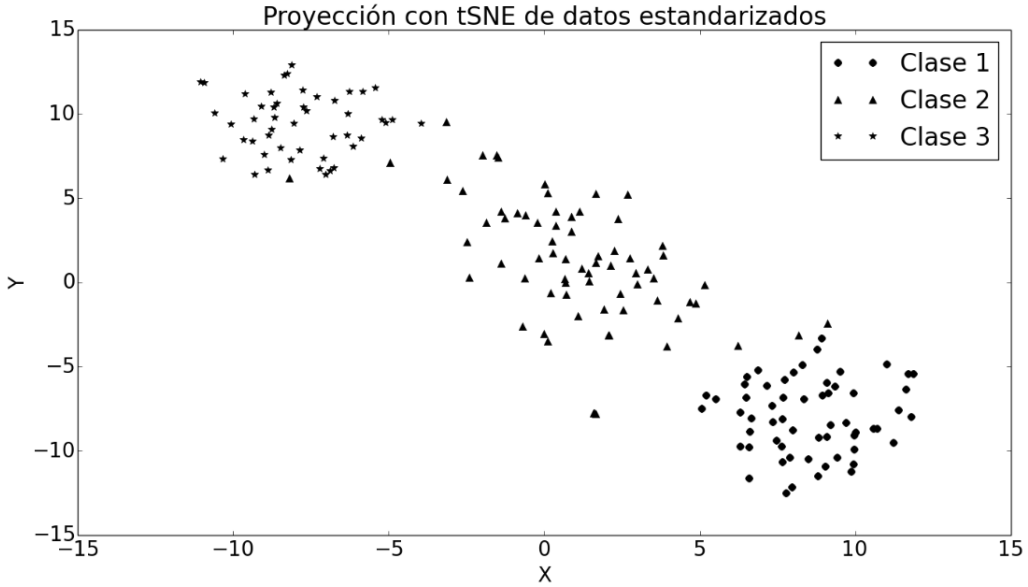


Figura 3.14: Proyección de datos estandarizados de la base de vinos.

3.5.3. Cáncer de pecho (Wisconsin)

Esta base de datos fue creada por el Dr. Wolberg durante el periodo de enero de 1989 a noviembre de 1991 muestreando y capturando sus casos clínicos [39], esta información fue divida en grupos para tener un orden cronológico en la adquisición de los datos. La base contiene 699 ejemplos de nueve características extraídas de tumores malignos y benignos para encontrar una relación entre las características morfológicas y el tipo de tumor encontrado, entre los atributos extraídos se encuentran: el espesor del tumor, uniformidad del tamaño de la célula, uniformidad en la forma de la célula, entre otros.

Este problema de clasificación es binario, ya que se desea clasificar si el tumor es maligno o benigno con base en las características elegidas. La complejidad está en la dimensionalidad de los datos, ya que son nueve atributos por cada vector. Para el entrenamiento se utilizó una red de 7 x 7 neuronas con topología hexagonal, además se utilizaron los datos sin procesar y después se normalizaron de forma global utilizando la técnica de mínimo-máximo.

3.5.4. Ionosfera

La siguiente base de datos fue creada con los datos obtenidos en Goose Bay por un radar, el cual consiste en arreglos de antenas de alta frecuencia. El objetivo era detectar estructuras de electrones libres en la ionosfera por medio de la medición de 34 atributos continuos, procesados por medio de la función de auto correlación [40]. La base de datos esta constituida por 351 datos con 34 características por cada uno, además se requirió utilizar técnicas para completar datos faltantes como el promedio de los dos datos anteriores ya que algunos datos no habían sido bien capturados para algunas características.

El problema de clasificación es binario y busca detectar si existen o no estructuras en la ionosfera usando el paso de las señales electromagnéticas por los electrones libres existentes en esta capa. La complejidad del problema radica en la alta dimensionalidad de los datos, ya que son señales electromagnéticas de 34 atributos cada una. Para el

entrenamiento se utilizó una red de 7 x 7 neuronas con topología hexagonal, así también se utilizaron los datos sin procesar y después se normalizaron de forma global utilizando la técnica del mínimo-máximo.

3.5.5. Semillas (seeds)

La base de datos fue creada para tareas de clasificación o agrupamiento. Se analizaron un grupo de granos pertenecientes a tres variedades de trigo: Kama, Rosa y Canadian, elegidos de forma aleatoria cada uno. Para la clasificación de estos tres grupos se midieron 7 atributos geométricos de los granos: área, perímetro, compacidad, longitud de grano, ancho de grano, coeficiente de asimetría y la longitud de la ranura del grano.

El problema de clasificación es multi-clase (tres clases) ya que se busca clasificar granos en tres familias distintas. Las clases se encuentran balanceadas y la base de datos está compuesta de 210 datos de 7 atributos cada uno. Para el entrenamiento se utilizó una red de 7 x 7 neuronas con topología hexagonal, además se utilizaron datos sin procesar y después se normalizaron de forma global utilizando la técnica de mínimo-máximo.

3.5.6. Datos artificiales bidimensionales

Los datos de esta base de datos fueron generados de forma artificial para realizar pruebas de clasificación con nuevos algoritmos [41]. La base está compuesta de 5000 datos de dos dimensiones, en la cual se observan los diferentes grupos que se forman en cúmulos bidimensionales, por otra parte las clases se encuentran balanceadas. El problema de clasificación es multi-clase, ya que son quince grupos diferentes que varían en forma: alargada, circular, elíptica, compacta y dispersa. Para clasificar la información se utilizó una red de 40 x 40 neuronas con topología hexagonal. Esta base de datos es caracterizada por los elementos que se encuentran en la frontera de cada grupo, ya que algunos datos son difíciles de clasificar por su cercanía a otros grupos. En la Figura 3.15 se observa la base de datos en donde cada color representa a un grupo diferente.

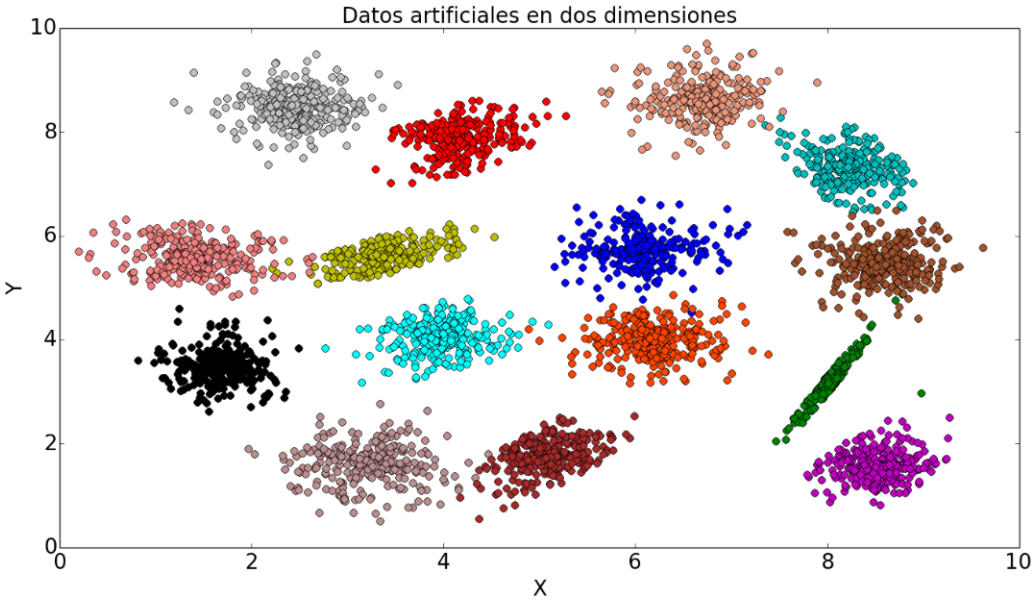


Figura 3.15: Base de datos de prueba bidimensional.

3.5.7. Datos artificiales tridimensionales

Los datos fueron generados de forma artificial para probar algoritmos de agrupamiento y clasificación. La base está compuesta de 6000 ejemplos de tres dimensiones, además las clases se encuentran balanceadas. El problema de clasificación es multi-clase, ya que son seis grupos diferentes que varían en forma, no obstante todos los cúmulos generados son muy compactos y bien separados entre ellos, marcando una división muy notoria entre grupos. Para clasificar la información se utilizó una red de 40 x 40 neuronas con topología hexagonal, la base está diseñada para que la mayoría de los datos sean correctamente clasificados ya que éstos no se encuentran dispersos en sus fronteras, en la Figura 3.16 se observan los datos en tercera dimensión agrupados en seis cúmulos bien separados.

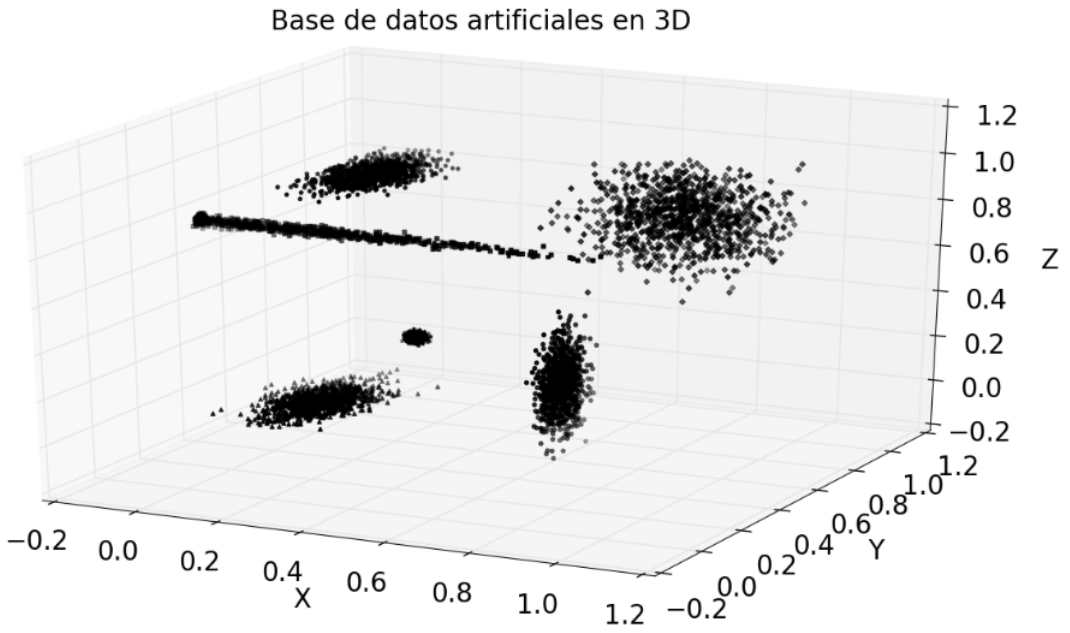


Figura 3.16: Base de datos de prueba tridimensional.

3.5.8. Dígitos escritos a mano (Semeion)

La base de datos esta constituida por dígitos escritos a mano por aproximadamente 80 personas, los cuales fueron escaneados en cuadrados de 16 x 16 píxeles en escala de grises de 256 valores. Cada píxel fue escalado en un valor booleano entre 0 y 1 utilizando un margen de umbral establecido [42]. Durante la captura de la información cada persona tuvo que escribir los dígitos del 0 al 9 dos veces, la primera vez de forma lenta cuidando los detalles y la segunda de forma rápida sin cuidar los detalles.

El problema de clasificación es multi-clase, ya que son diez dígitos los que se capturaron, el número de datos registrados son 1593, los cuales fueron segmentados siguiendo la metodología descrita en el pre-procesamiento de la base de datos sísmicos en donde se concatenaron las filas para formar los vectores de entrenamiento de 256 características, equivalente a todos los píxeles que componen cada imagen. Para el proceso de entrenamiento se utilizó una red de 40 x 40 neuronas con topología hexagonal, en la Figura 3.17 se observan los dígitos escritos utilizados para entrenar al modelo.

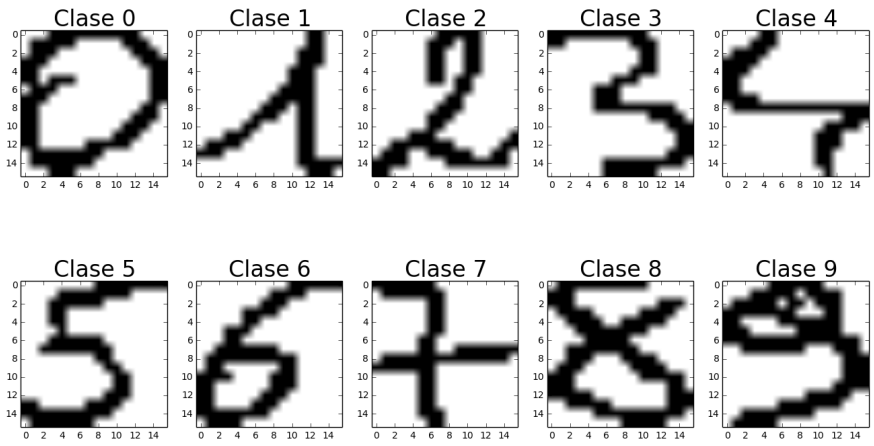


Figura 3.17: Dígitos de la base de datos semeion utilizados en el entrenamiento del modelo.

3.5.9. Datos sísmicos

En el capítulo anterior la base de datos recibió un pre-procesamiento para posteriormente ser analizada con la metodología por etapas. Como ya se mencionó, la base consta de nueve secciones de 460 señales con 625 muestras cada una. Primero, se realizaron diez entrenamientos con los mapas auto-organizados para elegir el de menor error topográfico. Una vez elegido el mapa se realizó un análisis de la calidad del agrupamiento por medio de índices de Davies Bouldin, Silhoutte y de representación múltiple.

Después del análisis por índices se seleccionó el número de clases en las cuales se dividió el mapa de neuronas para clasificar los datos. Una vez segmentado el mapa se comienza con el proceso de clasificación calculando la distancia euclidiana del vector de entrada y todas las neuronas clasificadas, asignándole la clase de la neurona más símil al vector. El proceso es iterativo hasta clasificar todas las zonas de las regiones sísmicas. En la Figura 3.18 se muestra un diagrama de flujo del proceso de clasificación.

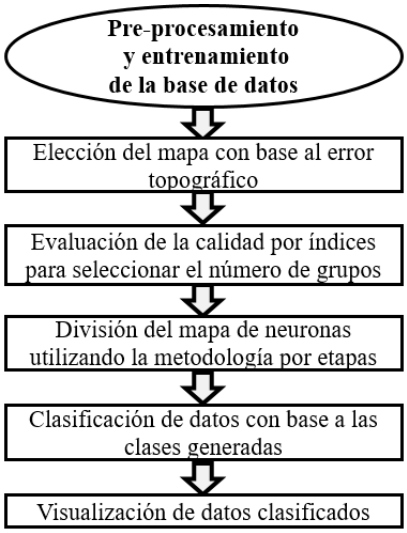


Figura 3.18: Diagrama de flujo del proceso de clasificación de datos sísmicos.

Los datos fueron clasificados con base a la forma de la señal, es decir, los cambios en amplitud de las trazas sísmicas, para esto se realizaron dos análisis: con los datos originales y con los datos en potencia. Ambos análisis permiten observar diferentes aspectos de la amplitud de onda de las señales.

La clasificación de datos sísmicos resulta compleja, ya que no existe un número de clases determinado para clasificar los datos. Por lo tanto, es importante seguir la metodología descrita y realizar primero una agrupación y creación de clases con los datos de entrenamiento, para posteriormente clasificar la información restante que no ha sido utilizada. Para este proceso de entrenamiento se utilizó una red de 40 x 40 neuronas con topología hexagonal.

Capítulo 4

Resultados

En este capítulo se muestran los resultados obtenidos después de realizar las pruebas de clasificación descritas en el capítulo anterior. Primero, se describen los resultados obtenidos para las pruebas con las bases de datos etiquetadas, empleando diferentes métricas de evaluación para calificar el desempeño del modelo al clasificar datos no utilizados durante la creación de las clases, además se abordan diferentes etapas de desarrollo del modelo. Después se muestra el proceso para clasificar los datos sísmicos, ya que a diferencia de las pruebas anteriores estos no cuentan con etiquetas para evaluar el desempeño.

4.1. Bases de datos etiquetadas

Todas las pruebas se realizaron con una configuración estándar, es importante señalar que estos parámetros se fijaron para comparar el desempeño obtenido del modelo ante diferentes tipos de datos, es decir, probar el modelo con datos de diferentes bases de datos. En la Tabla 4.1 se muestran características de las bases de datos y parámetros del modelo.

Tabla 4.1: Parámetros de configuración para entrenar con los mapas auto-organizados y características de las bases de datos.

Base	Tamaño (mapa)	Atributos	Épocas	Vecindad
Iris	4x4	3	150000	0.4
Vinos	4x4	13	178000	0.4
Cáncer	7x7	9	699000	0.7
Ionosfera	7x7	34	351000	0.7
Semillas	7x7	7	210000	0.7
Datos 2D	40x40	2	5000000	4
Datos 3D	10x10	3	6000000	1
Semeion	40x40	256	1600000	4

Se empleó una validación cruzada *K-Fold* con K = 10, tomando el 80 % de datos para el entrenar y 20 % para validar y calificar el desempeño del modelo. Durante el entrenamiento para cada k iteración se utilizó el mismo porcentaje de división para entrenar y validar con los datos de entrenamiento, además la selección de los datos que conforman a los dos conjuntos fueron generados aleatoriamente para cada k iteración.

4.1.1. Iris

En las primeras pruebas con el modelo se utilizó una topología de red cuadricular, es decir, las neuronas tenían cuatro neuronas vecinas y las relaciones diagonales entre ellas no tenían influencia en la distribución de los datos, en la Figura 4.1 se observa un mapa entrenado con esta topología de red.

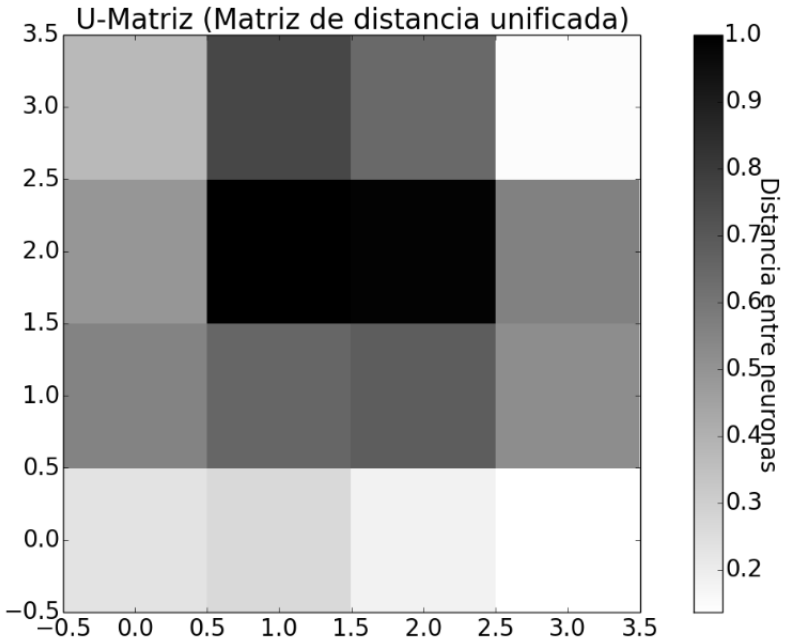


Figura 4.1: Mapa auto-organizado entrenado con topología cuadricular utilizando la base de datos de iris.

Con este tipo de topología se obtuvieron resultados en clasificación de máximo 80 % de precisión con una red de tamaño 40 x 40 para la base de datos iris, esto causado debido a la nula conectividad entre las diagonales de las neuronas que afectó la distribución de los datos en el mapa, especialmente los que se encontraban en la frontera entre dos grupos, ya que la clasificación era errónea entre ellos. Por lo tanto, se optó por utilizar la topología hexagonal en todas las pruebas, evitando así los problemas de distribución de los datos durante el entrenamiento.

El entrenamiento de la red neuronal se realizó con datos normalizados y sin normalizar, para ambos casos después de las k iteraciones se eligió al mapa con el error topográfico menor. Se seleccionó esa métrica de error ya que esta relacionada con la topología de los datos e indica cuando ocurren plegamientos de la red de neuronas. En la Tabla 4.2 se muestran los resultados obtenidos después de evaluar al modelo con la validación cruzada de Monte Carlo, donde S/N se refiere a los datos sin normalizar y N a los datos normalizados.

Tabla 4.2: Medición del desempeño del modelo con la base de datos de iris.

Métrica de evaluación	Evaluación S/N	Evaluación N
Error topográfico	0.05	0.05
Error de cuantización	0.03	0.03
Precisión con K=10	93 %	96 %
Exactitud	91 %	91 %
Precisión	92 %	92 %
Sensibilidad	92 %	92 %
Especificidad	95 %	95 %
Valor F1	92 %	92 %

Se observa en la Tabla 4.2 que las métricas de evaluación oscilan entre 91 % - 96 %, lo que quiere decir que el modelo aprendió y generalizó de buena forma obteniendo un porcentaje de 92 % de precisión a la hora de clasificar los datos de validación. Además, el modelo fue capaz de identificar bien a los ejemplos que fueron correctamente clasificados con una sensibilidad del 92 % y también a los datos que no fueron correctamente clasificados con una especificidad del 95 %. Por otra parte, en las Figuras 4.2 y 4.3 se observa la división del mapa de neuronas en grupos durante las tres etapas descritas en el capítulo 3, cada color representa a un grupo diferente durante las etapas.

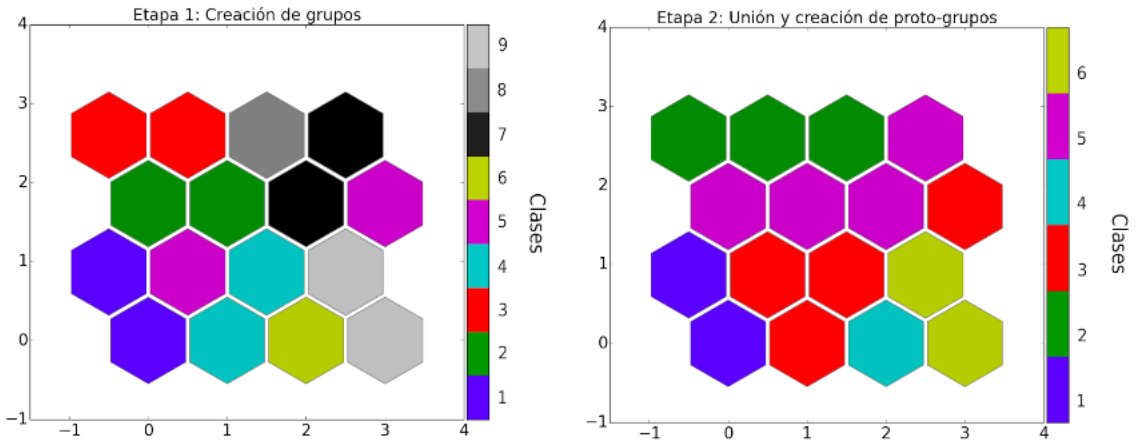


Figura 4.2: Etapa uno y dos del proceso de creación de clases.

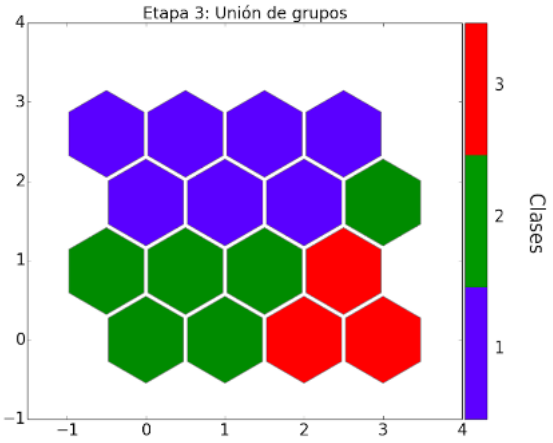


Figura 4.3: Etapa tres del proceso de creación de clases.

4.1.2. Vinos

Las pruebas para la base de datos de vinos se realizaron con los datos sin procesar y posteriormente estandarizados. Después de las k iteraciones se eligió el mapa con el error topográfico menor para ser utilizado en el modelo. En la Tabla 4.3 se muestran los resultados obtenidos después de evaluar al modelo con la validación cruzada de Monte Carlo, donde S/E se refiere a los datos sin estandarizar y E a los datos estandarizados.

Tabla 4.3: Medición del desempeño del modelo con la base de datos de vinos.

Métrica de evaluación	Evaluación S/E	Evaluación E
Error topográfico	0.14	0.14
Error de cuantización	24.06	1.82
Precisión con K=10	75 %	95 %
Exactitud	56 %	96 %
Precisión	57 %	95 %
Sensibilidad	57 %	95 %
Especificidad	76 %	96 %
Valor F1	57 %	95 %

En la Tabla 4.3 se observa que las métricas de evaluación entre los datos estandarizados y no estandarizados difieren mucho, pues como se describió en el Capítulo 3 la estandarización ayudó a la separación de los datos que no estaban a la misma escala, además de que existen fuertes dependencias lineales en dos grupos que se traslapan dificultando la clasificación. Así mismo, se observa que el error de cuantización para los datos no estandarizados es muy superior al de los datos estandarizados, debido a que los patrones aprendidos no fueron buenos y por ende el ajuste del modelo tampoco.

Por otra parte, después de la estandarización, el modelo mostró resultados entre 94 % - 96 % en las métricas de evaluación, obteniendo un 95 % en la exactitud al momento de clasificar los datos de validación. También se muestra que el modelo identifica bien los ejemplos correctamente clasificados y de igual forma los que no fueron correctamente clasificados. Además, las medidas de error del entrenamiento de los mapas se mantienen bajas, lo que indica un correcto ajuste de los datos y una buena distribución de éstos al momento del mapeo.

4.1.3. Cáncer de pecho (Wisconsin)

La red neuronal para la base de datos fue entrenada de dos maneras: con los datos normalizados y sin normalizar. En la Tabla 4.4 se muestran los resultados obtenidos después de evaluar al modelo con la validación cruzada de Monte Carlo, donde S/N se refiere a los datos sin normalizar y N a los datos normalizados.

Tabla 4.4: Medición del desempeño del modelo con la base de datos de cáncer de pecho.

Métrica de evaluación	Evaluación S/N	Evaluación N
Error topográfico	0.27	0.24
Error de cuantización	2.24	0.25
Precisión con K=10	95 %	95 %
Exactitud	98 %	98 %
Precisión	98 %	98 %
Sensibilidad	98 %	98 %
Especificidad	94 %	94 %
Valor F1	98 %	98 %

En la Tabla 4.4 se observa que el desempeño después de la normalización de los datos no varía mucho, el modelo aprendió y generalizó la estructura de los datos de entrenamiento ya que el modelo logró un desempeño del 98 % en precisión al clasificar de forma correcta los datos, de igual manera identifica de buena forma los datos correctamente clasificados y también los que no fueron correctamente clasificados. Por otra parte, las medidas de error para evaluar el entrenamiento del mapa indican que los datos se distribuyeron bien en el mapa y además que el ajuste del modelo fue correcto.

Debido a que la base de datos tiene ejemplos con una dimensionalidad alta existen herramientas como la *u-matriz*, que permiten observar en dos dimensiones las estructuras formadas por los datos al momento de mapear los datos de entrenamiento al mapa. En la Figura 4.4 se observa la matriz de distancia unificada después del entrenamiento y también el mapa de neuronas luego de ser dividido en dos clases para la clasificación

de los datos.

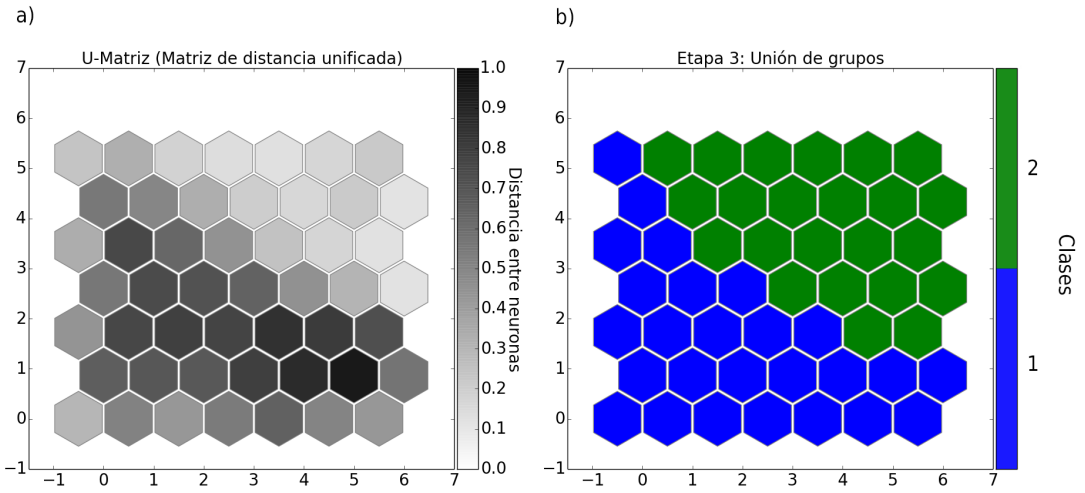


Figura 4.4: a) Matriz de distancia unificada b) Mapa de neuronas dividido en clases.

En la Figura 4.4 se observa que la *u-matriz* presenta un cambio en la intensidad de color a una tonalidad obscura, por la parte media del mapa hacia las neuronas de la parte inferior, indicando una distancia mayor entre las neuronas en esa zona, que coincide con la división creada por el modelo en el mapa de neuronas. Este cambio en la intensidad indica la separación en dos clases de los datos de entrenamiento mapeados en la red y además que se esta conservando la topología de los datos mapeados al crear las clases en el mapa de neuronas. Así también se muestra que las medidas de desempeño del entrenamiento indican un ajuste correcto de los datos y una buena distribución en el mapeo.

4.1.4. Ionosfera

Las pruebas con la red neuronal se realizaron normalizando los datos utilizando la técnica del mínimo-máximo y con los datos sin normalizar. En la Tabla 4.5 se muestran los resultados obtenidos después de evaluar el modelo con la validación cruzada de Monte Carlo, donde S/N se refiere a los datos sin normalizar y N a los datos normalizados.

Tabla 4.5: Medición del desempeño del modelo con la base de la ionosfera.

Métrica de evaluación	Evaluación S/N	Evaluación N
Error topográfico	0.26	0.14
Error de cuantización	1.26	0.63
Precisión con K=10	73 %	92 %
Exactitud	71 %	87 %
Precisión	71 %	87 %
Sensibilidad	71 %	87 %
Especificidad	78 %	85 %
Valor F1	71 %	87 %

En la Tabla 4.5 se observa que las métricas de evaluación del desempeño varían entre cinco a quince puntos porcentuales, lo cual es una cantidad considerable cuando se quiere garantizar que el modelo ha generalizado de forma correcta la estructura de los datos. Cuando los datos no son normalizados se cuenta con un porcentaje en promedio del 72 % en todas las medidas de evaluación, lo que significa que un cuarto de los datos no son clasificados de forma correcta y de igual manera no se está identificando correctamente las clasificaciones.

Cuando los datos son normalizados se observa una notable mejoría en las medidas de evaluación del modelo, es decir, el modelo generaliza mejor porque la precisión incrementa así como la sensibilidad y la especificidad entre 85 % a 87 % en las métricas de evaluación. Por otra parte, las métricas de error para evaluar el entrenamiento del mapa disminuye, sin embargo, algunos datos del entrenamiento no fueron mapeados de forma correcta.

Debido a que la base tiene datos con estructuras complejas (34 atributos), existen técnicas para visualización de información en alta dimensión como tSNE que realiza una proyección de estos a un espacio bidimensional, para visualizar las estructuras que forman los vectores. En la Figuras 4.5 y 4.6 se observan las proyecciones con tSNE

antes y después de la normalización de los datos.

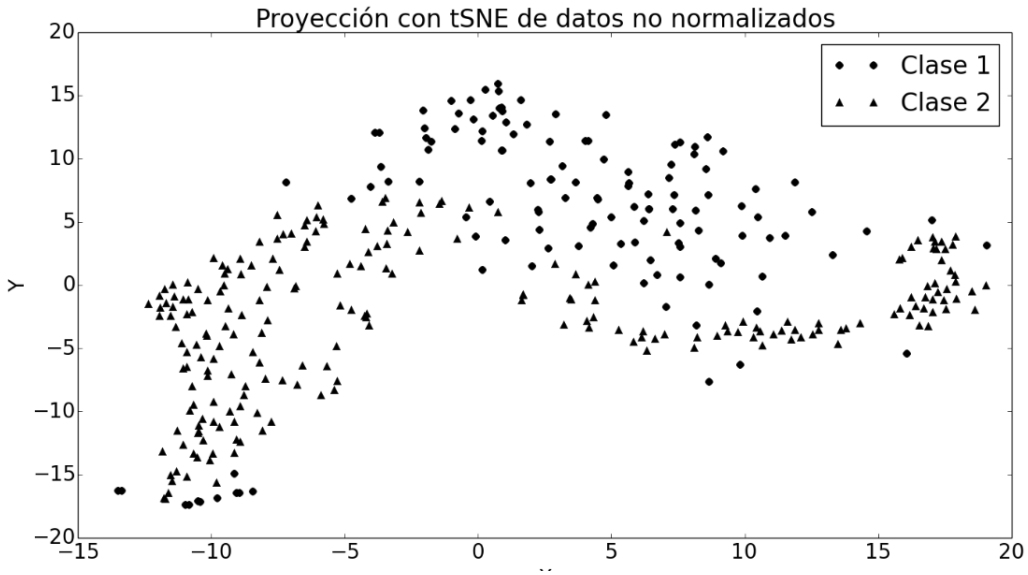


Figura 4.5: Datos sin normalizar de la base de datos de la ionosfera.

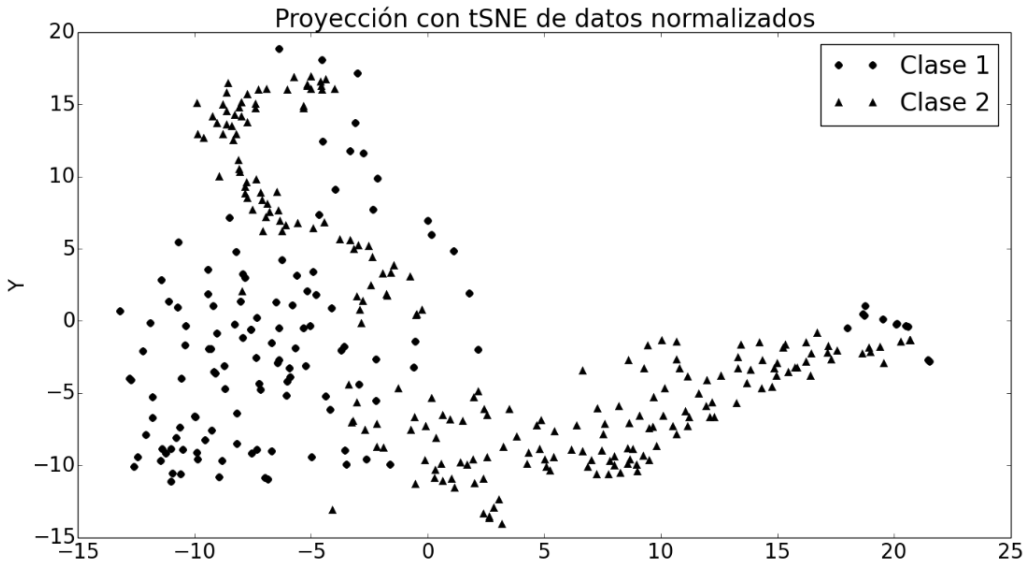


Figura 4.6: Datos normalizados de la base de datos de la ionosfera.

En las Figuras 4.5 y 4.6 se observa que los datos sin normalizar se encuentran separados en dos cúmulos y que algunos ejemplos se traslapan entre clases. Por el

contrario cuando estos son normalizados la clase dos se separa más de la clase uno traslapando algunos ejemplos de ambas clases, pero sin que exista partición del grupo como en la Figura 4.5. Si bien la normalización ayudó a separar un poco más los datos como se observa en la proyección, la alta dimensionalidad del problema dificulta la clasificación porque existen relaciones entre los datos que no es trivial observar. En el anexo A, se muestra la división del mapa de neuronas por etapas.

4.1.5. Semillas

El mapa fue entrenado con datos normalizados por medio de la técnica del mínimo-máximo y con datos sin normalizar. En la Tabla 4.6 se muestran los resultados obtenidos después de evaluar el modelo con la validación cruzada de Monte Carlo, donde S/N se refiere a los datos sin normalizar y N a los datos normalizados.

Tabla 4.6: Medición del desempeño del modelo con la base de semillas.

Métrica de evaluación	Evaluación S/N	Evaluación N
Error topográfico	0.14	0.22
Error de cuantización	0.36	0.02
Precisión con K=10	90 %	89 %
Exactitud	90 %	91 %
Precisión	90 %	90 %
Sensibilidad	90 %	90 %
Especificidad	92 %	93 %
Valor F1	90 %	90 %

En la Tabla 4.6 las medidas de evaluación utilizadas con los datos normalizados y sin normalizar no varían mucho, por lo que se observa el modelo generalizó bien la estructura y se ajusto de buena manera a los datos teniendo resultados de precisión del 90 % en clasificación, de igual forma el porcentaje en la identificación de los datos correctamente clasificados y los que no fueron correctamente clasificados es alto. Por

otra parte, las medidas de error para evaluar el entrenamiento del mapa disminuyen entre los datos normalizados y los no normalizados pero no influyen mucho para mejorar la tasa clasificación de datos. En el anexo A, se muestra la división del mapa de neuronas por etapas.

4.1.6. Datos artificiales bidimensionales

La base de datos artificiales bidimensional fue utilizada para analizar el comportamiento del modelo ante una cantidad de datos mayor, ya que las pruebas anteriores no excedía a los 700 datos por cada base. Las características importantes de estos datos, son la forma espacial de los grupos en el plano cartesiano, así como la cantidad de grupos en los que se dividió la información (quince). La red neuronal se entrenó con datos normalizados mediante la técnica del mínimo-máximo y sin normalizar.

En las Figuras 4.7 y 4.8 se observan las *u-matriz* de los datos antes y después del entrenamiento. En la etapa previa al entrenamiento no se muestra una estructura en la red neuronal ya que se inicializa con pesos aleatorios, no obstante, después de realizado el entrenamiento en la Figura 4.8 se observan regiones creadas en el mapa separadas por zonas obscuras que indican grandes cambios de distancia entre neuronas.

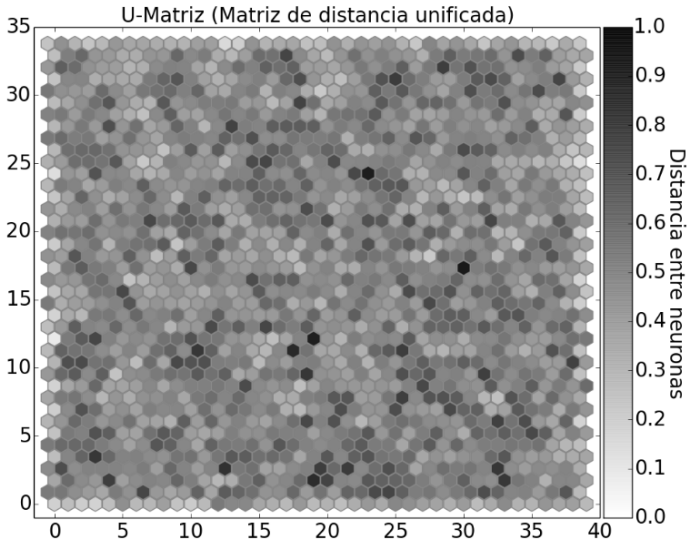


Figura 4.7: U-matriz de los datos artificiales antes del entrenamiento.

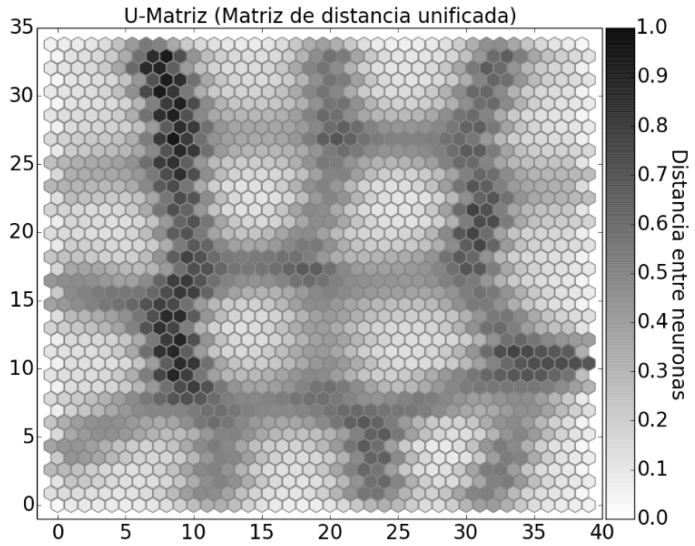


Figura 4.8: U-matriz de los datos artificiales después del entrenamiento.

En la Tabla 4.7 se muestran los resultados obtenidos después de evaluar el modelo con la validación cruzada de Monte Carlo, donde S/N se refiere a los datos sin normalizar y N a los datos normalizados.

Tabla 4.7: Medición del desempeño del modelo con la base de datos artificiales en 2D.

Métrica de evaluación	Evaluación S/N	Evaluación N
Error topográfico	0.05	0.03
Error de cuantización	0.08	0.04
Precisión con K=10	94 %	98 %
Exactitud	95 %	97 %
Precisión	95 %	97 %
Sensibilidad	95 %	97 %
Especificidad	96 %	98 %
Valor F1	95 %	97 %

En la Tabla 4.7 se observa que las medidas de evaluación utilizando los datos sin normalizar y los normalizados no varían mucho, pero el modelo se ajusto bien a los datos

ya que al clasificar tiene una precisión promedio del 96 % confundiéndose un poco al momento de clasificar la clase 3 y la 4 por la delimitación de sus fronteras en el mapa de neuronas, de igual forma el porcentaje de identificación de los datos correctamente clasificados y los que no fueron correctamente clasificados es alto. Por otra parte, las medidas de error que evalúan el mapa indican una distribución correcta de los datos en todo el mapa ya que el error es mínimo.

Los resultados obtenidos en la clasificación a partir de las métricas de evaluación se pueden observar de forma gráfica en las Figuras 4.9 y 4.10, donde se muestran el mapa de neuronas dividido en las quince clases, cada clase codificada con un color diferente y también los datos artificiales coloreados por la clase a la que fueron asignados.

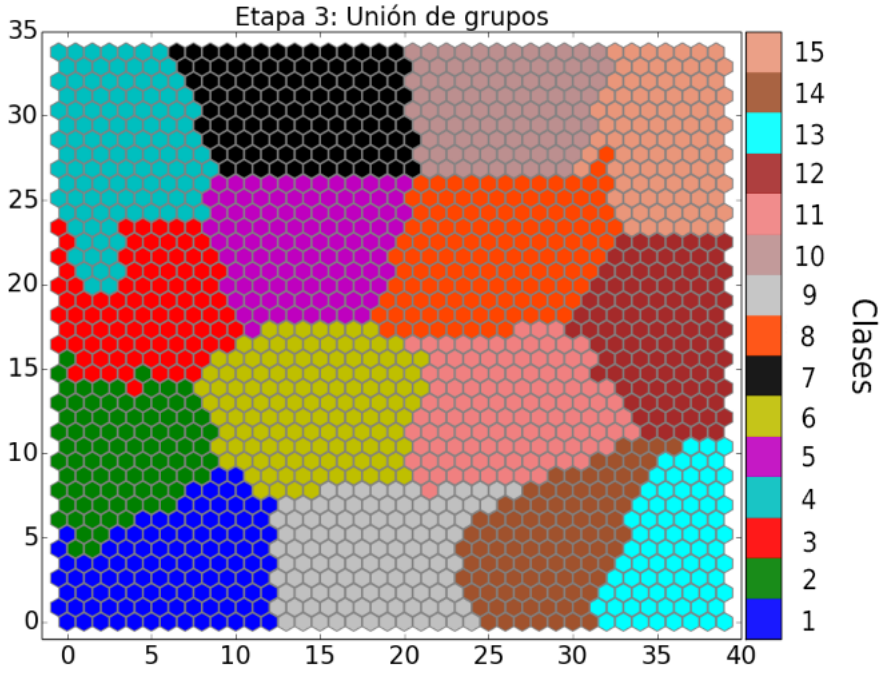


Figura 4.9: Mapa de neuronas dividido por clases.

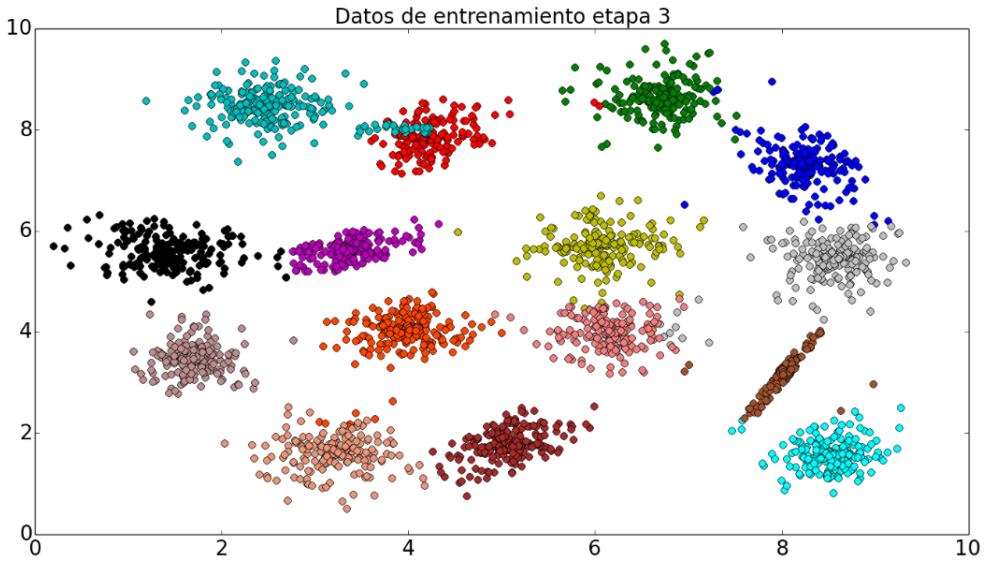


Figura 4.10: Datos clasificados con base al color de la clase asignada en el mapa de neuronas.

En la Figura 4.9 y 4.8 se observa que el modelo dividió al mapa de neuronas conservando la topología mapeada durante el entrenamiento del mapa auto-organizado que se observa en la *u-matriz*, es decir, se preservó en ambas etapas la topología de los datos. Por otra parte, en la Figura 4.10 se muestran los datos clasificados en diferentes grupos diferenciados por el color de la clase, además durante la división del mapa de neuronas se muestra que la clase cuatro se extendió más de lo que debería sobre la clase tres con base en la topología mostrada en la *u-matriz*, lo que generó que datos de ambas clases sean clasificados de forma errónea. Así también existen clasificaciones erróneas en los datos que se encuentran en las fronteras de los grupos, sobre todo en los grupos que tienen formas no tan compactas. En el Anexo A, se muestran todas las etapas de la división del mapa de neuronas y la clasificación de los datos durante las tres etapas.

4.1.7. Datos artificiales tridimensionales

La base de datos artificiales de tres atributos se utilizó como base de datos control para analizar el proceso de creación de grupos durante las tres etapas de división del

mapa de neuronas, fue elegida como control ya que son seis clases bien separadas y compactas. La red neuronal se entrenó con los datos sin normalizar. En la Tabla 4.7 se muestran los resultados obtenidos después de evaluar el modelo con la validación cruzada de Monte Carlo.

Tabla 4.8: Medición del desempeño del modelo con la base de datos artificiales en 3D.

Métrica de evaluación	Evaluación SN
Error topográfico	0.04
Error de cuantización	0.04
Precisión con K=10	100 %
Exactitud	100 %
Precisión	100 %
Sensibilidad	100 %
Especificidad	100 %
Valor F1	100 %

En la Tabla 4.8 se observa que en todas las métricas el resultados es del 100 %, es decir, el modelo generalizó correctamente la estructura de los datos logrando clasificarlos correctamente a todos, además se identificó a todos los ejemplos clasificados. Por otra parte, las medidas de error para evaluar el mapa indican que los datos fueron bien distribuidos por el mapa durante el entrenamiento.

Los resultados obtenidos de la clasificación de los datos en seis clases se observan en las Figuras 4.11 y 4.12, en donde se muestran el mapa de neuronas dividido en las seis clases, cada una codificada con un color diferente, además existen dos clases extras que son neuronas sin asignar ya que nunca fueron activadas durante el entrenamiento. Además, se muestran igual los datos artificiales coloreados por la clase a la que fueron asignados. En el Anexo A, se muestran todas las etapas de la división del mapa de neuronas y el proceso de clasificación de los datos durante las tres etapas.

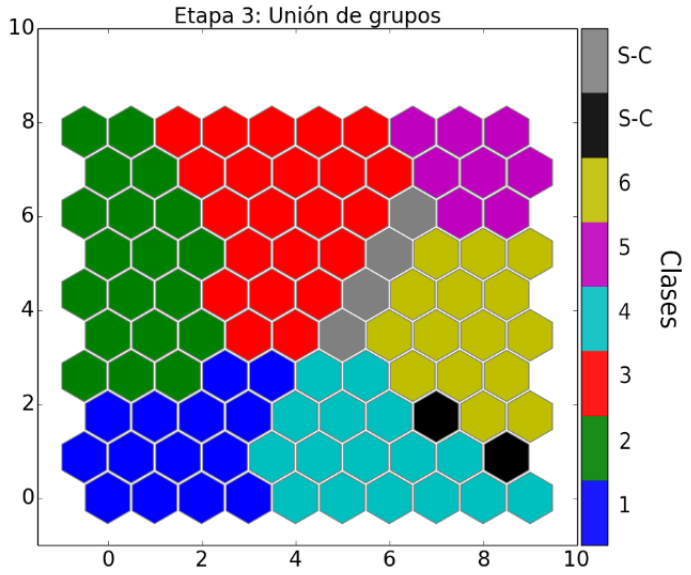


Figura 4.11: Mapa de neuronas dividido por clases.

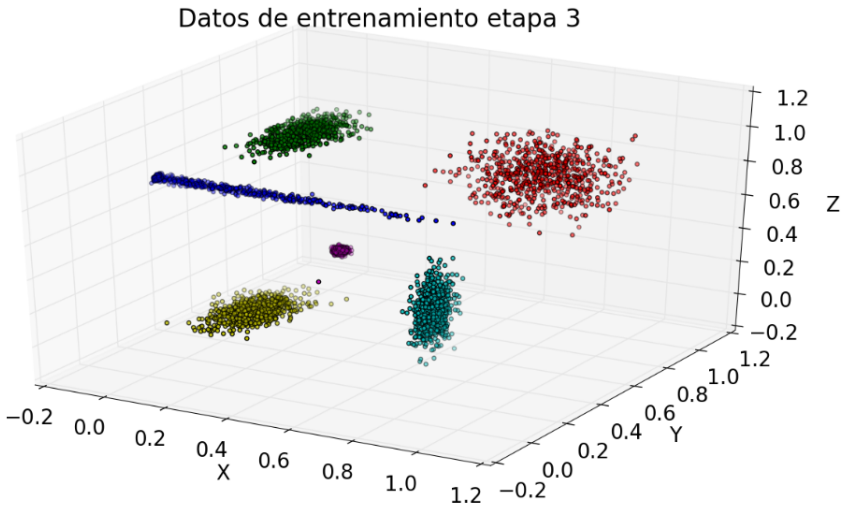


Figura 4.12: Datos clasificados con base al color de la clase asignada en el mapa de neuronas.

4.1.8. Dígitos escritos a mano (Semeion)

Esta base de datos compuesta de 1593 imágenes de 16 x 16 píxeles de los dígitos del 0 al 9 cuenta con un pre-procesamiento, las imágenes fueron binarizadas utilizando un umbral para transformar los valores a 0's y 1's [42]. En la Figura 4.13 se observa la proyección de estos vectores de alta dimensión en dos dimensiones utilizando la técnica tSNE.

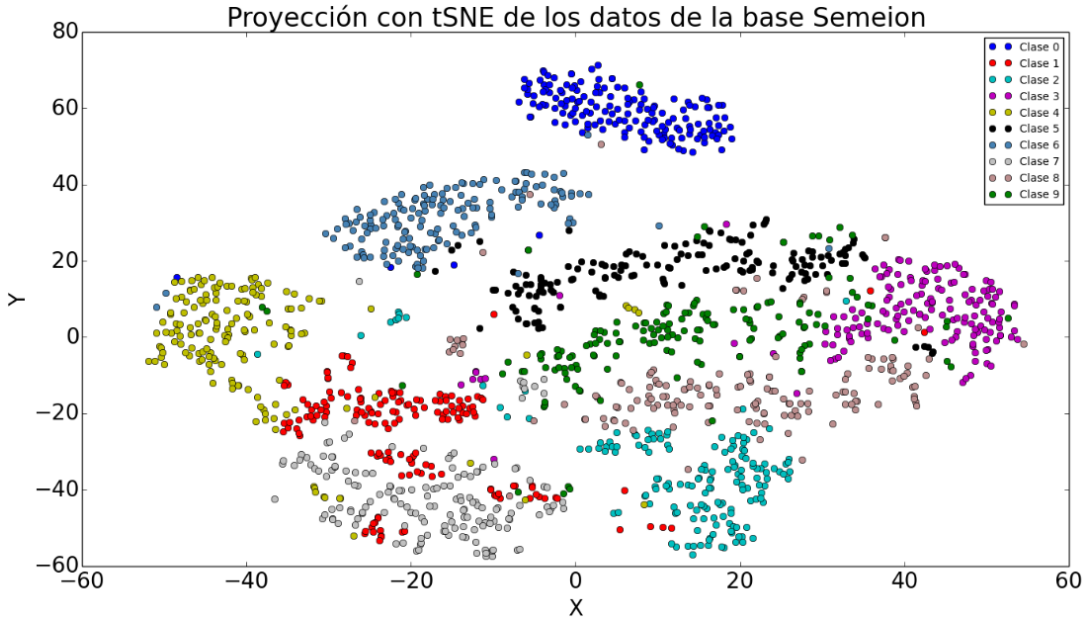


Figura 4.13: Proyección de los datos en dos dimensiones usando tSNE.

En la Figura 4.13 se observa que las clases 0, 7 y 4 son las que menos se traslapan con las otras, la complejidad de los datos dificulta la observación de las estructuras formadas por ellos, sin embargo, con ayuda de herramientas de visualización de datos en alta dimensión se genera una aproximación de su distribución en un plano bidimensional. El proceso de entrenamiento se realizó con los datos transformados en vectores de entrenamiento, en la Figura 4.14 se observan los coeficientes transformados en imágenes y ordenados en la forma del mapa de neuronas para observar las dígitos mapeados después del entrenamiento.

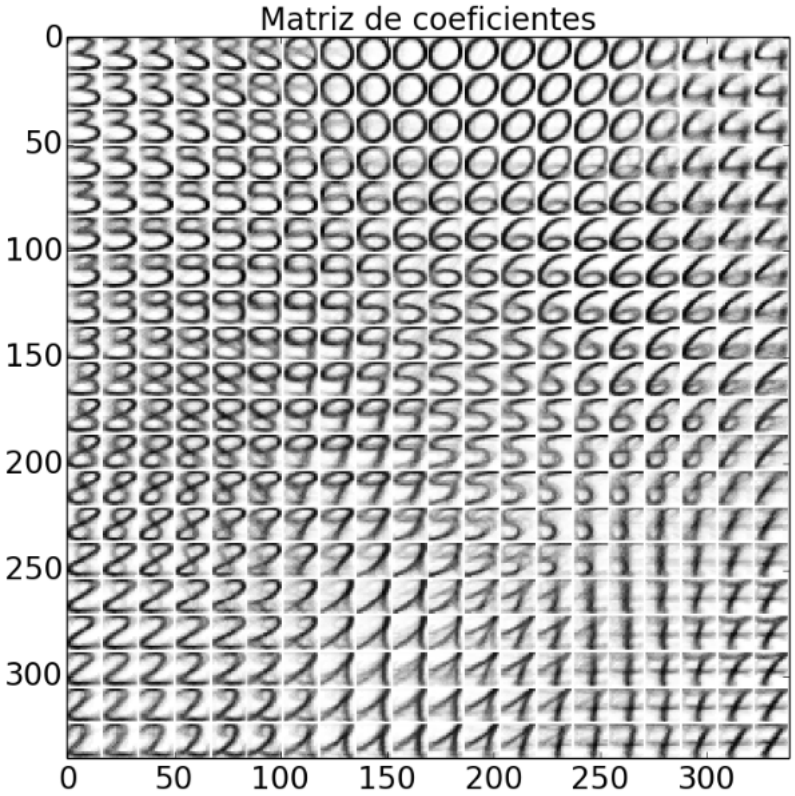


Figura 4.14: Coeficientes de la neuronas transformados a imágenes después del entrenamiento del mapa.

En la matriz de coeficientes se muestra que el mapa ha aprendido las formas de los números del 0 al 9, sin embargo, existen zonas de transición entre las clases 1, 3, 5, 8 y 9 en las cuales los coeficientes muestran números borrosos entre clases, es decir, no se aprecian bien los dígitos en esas neuronas, lo que indica que esas zonas fueron mapeadas utilizando diferentes datos por cada ciclo causando que la transición entre clases no sea correcta. En la Tabla 4.9 se muestran los resultados obtenidos después de evaluarlo con la validación cruzada de Monte Carlo.

Tabla 4.9: Medición del desempeño del modelo con la base semeion.

Métrica de evaluación	Evaluación de los datos
Error topográfico	0
Error de cuantización	5.38
Precisión con K=10	61 %
Exactitud	63 %
Precisión	63 %
Sensibilidad	63 %
Especificidad	97 %
Valor F1	63 %

En la Tabla 4.9 se muestra el resultado de las métricas de evaluación, se observa que el modelo no generalizó bien la estructura de los datos, debido a los traslapes de ejemplos entre clases que se observan como números difuminados en la matriz de coeficientes. Además, el modelo no identificó bien a los ejemplos clasificados correctamente pero si a los que fueron clasificados de forma incorrecta como la clase 9. Así también, el error de cuantización indica que los patrones aprendidos por la red no son buenos y se aprecian en las fronteras de las clases en la matriz de coeficientes 4.14, por lo tanto, el ajuste del modelo no fue bueno. En relación a las clases que fueron clasificadas erróneamente se utilizó la matriz de confusión que se muestra en la Figura 4.15 para examinar a detalle donde ocurrieron estos errores de clasificación.

CLASE VERDADERA	Clase 0	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6	Clase 7	Clase 8	Clase 9
Clase 0	25	0	0	0	0	0	0	0	0	1
Clase 1	0	17	0	0	0	0	0	9	0	0
Clase 2	0	0	19	0	0	0	0	1	6	0
Clase 3	0	2	0	20	0	1	1	0	2	0
Clase 4	1	4	0	0	10	3	0	1	1	6
Clase 5	0	1	0	7	1	11	3	3	0	0
Clase 6	7	0	0	0	0	0	15	0	0	4
Clase 7	0	2	0	0	0	0	0	21	3	0
Clase 8	0	1	2	5	0	0	0	0	17	0
Clase 9	0	0	0	9	0	9	0	2	6	0

CLASE PREDICHA

Figura 4.15: Matriz de confusión de los dígitos del 0 al 9.

En la matriz 4.15 la diagonal marcada con rayas es donde la clase predicha es igual a la clase verdadera, en otras palabras, es donde la clasificación es correcta, así también los errores que sobrepasen a 3 datos entre clases son resaltados en la matriz. Se observa que la clase 9 no acertó con ningún dato pero se confunde con las clases 3, 5 y 8, por otra parte para la clase 6 existen problemas de clasificación con las clases 0 y 9. También se muestra que existen varios datos mal clasificados para la mayoría de clases a excepción de las clases 0 y 3 las cuales tienen muy pocos errores de clasificación. Para concluir el análisis de esta base de datos en la Figura 4.16 se muestra el mapa de neuronas dividido en las diez clases en donde dentro de cada neurona está indicado el dígito al que pertenece cada clase.

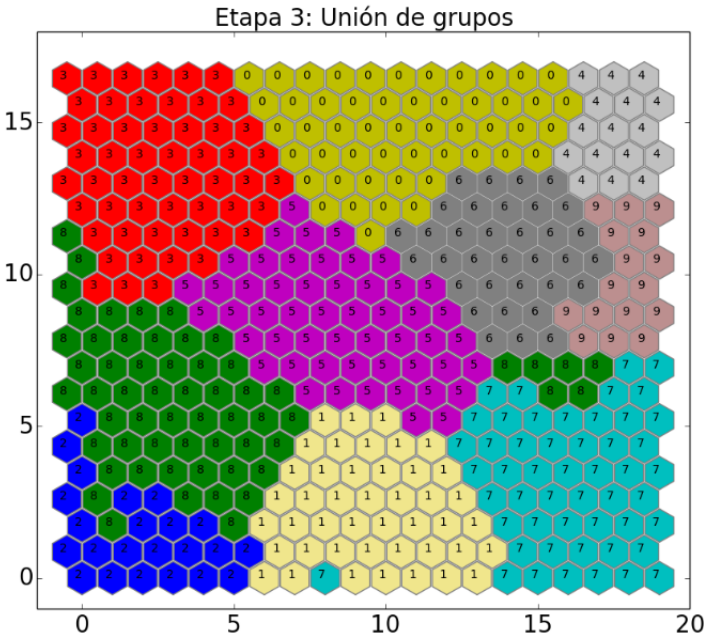


Figura 4.16: Mapa de neuronas dividido por clases.

El mapa de neuronas fue dividido conservando la topología que se observa en la matriz de coeficientes 4.14, es decir, la posición de las clases en el mapa coinciden con la división del mapa por lo que se detectó la ubicación espacial de las clases mapeadas, sin embargo, el dígito 9 se encuentra distribuido por el centro y presenta problemas de clasificación con las clases 3, 5, 6 y 8, además el mapeo de las neuronas fronteras no se realizaron de forma correcta.

4.2. Base de datos no etiquetada (Datos sísmicos)

Las primeras pruebas de agrupamiento para clasificar datos se llevaron a cabo utilizando la metodología 1, es decir, el agrupamiento por algoritmos genéticos, en la Figura 4.17 se observa el resultado del agrupamiento.

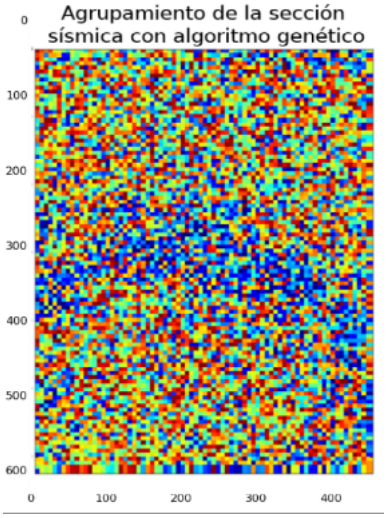


Figura 4.17: Resultado del agrupamiento con el algoritmo genético.

Como se observa en la Figura 4.17 el resultado del agrupamiento de la sección sísmica con el algoritmo genético no muestra regiones homogéneas y los patrones aprendidos no tienen sentido. Además, el tiempo de ejecución del proceso entre cada prueba era muy prolongado de aproximadamente cinco días, lo que resulto inviable para generar resultados rápidos y confiables. Entonces se continuo con la segunda metodología para clasificar.

Una vez que se seleccionaron y pre-procesaron los vectores de entrenamiento se procedió a realizar una serie de diez entrenamientos con mapas diferentes, eligiendo el mapa con menor error topográfico como se realizó con las bases de datos etiquetadas. Después, se inicio el proceso de evaluación de los agrupamientos mediante los índices. Para esto se utilizó el mapa seleccionado y se aplicó el proceso de división del mapa neuronal de forma iterativa, iniciando con una división de 10 grupos con aumento de uno en uno hasta 20 grupos. Los índices fueron evaluados para cada partición y se obtuvo una gráfica para seleccionar la cantidad de grupos que serían formados con los datos. Se utilizaron tres técnicas para evaluar la calidad, cada una basada en fundamentos diferentes pero manteniendo la idea de crear grupos bien separados y compactos.

Las primeras propuestas de clasificación de las secciones sísmicas fue utilizar a las

neuronas como las clases verdaderas en las que se clasificarían los datos, limitando el tamaño de los mapas a la cantidad de grupos en que se deseaba dividir la información, en la Figura 4.18 se observa una clasificación de la zona sísmica utilizando un mapa de 4 x 4 neuronas dividido en 16 clases distintas y clasificando cada dato de validación por medio de la evaluación de la distancia euclidiana con los coeficientes de cada neurona.

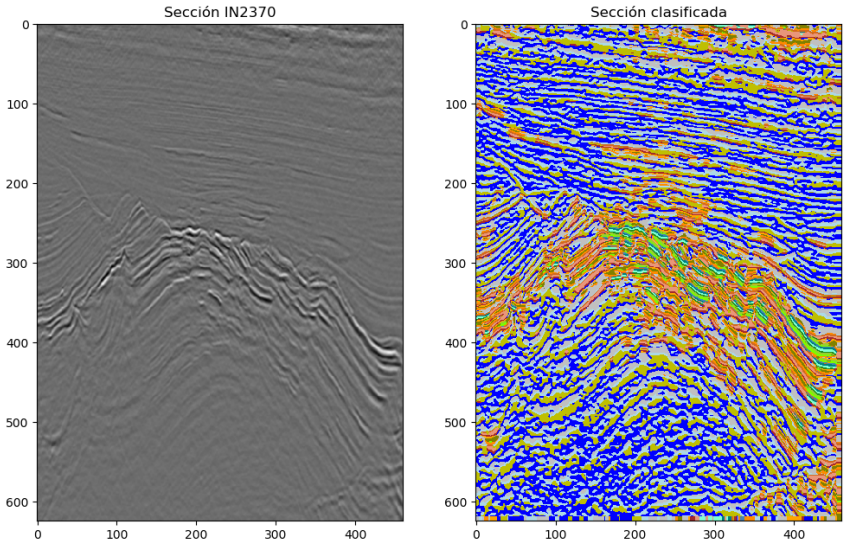


Figura 4.18: Clasificación de una sección sísmica en 16 clases diferentes.

Como se observa la clasificación de la sección sísmica fue realizada basada en cada neurona como grupo diferente, sin embargo, algunas zonas no muestran buenos resultados ya que las divisiones entre capas de material no se observan bien, lo que dificulta encontrar secciones homogéneas en la información. Al dividir el mapa de neuronas los grupos son representados de forma múltiple mediante diferentes neuronas, en la Figura 4.19 se muestra la evaluación grupo por grupo de las técnicas para seleccionar el número de clases (etiquetas).

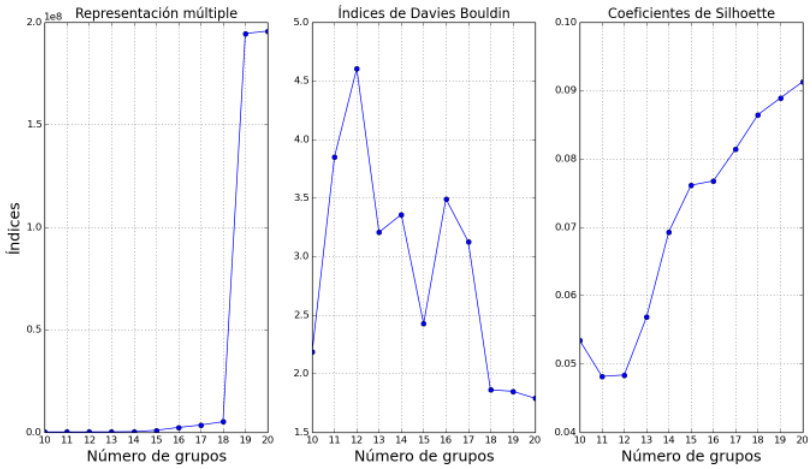


Figura 4.19: Evaluación de la calidad del agrupamiento utilizando las técnicas por índices descritas.

En la Figura 4.19 se observa que entre 18 a 20 grupos se obtienen las mejores evaluaciones en los tres índices, sin embargo, se optó por seleccionar 15 grupos, ya que es donde se presenta la primera caída abrupta en la evaluación por el índice de Davies Bouldin para la división del mapa de neuronas. Después de la división se clasificaron todas las regiones sísmicas basadas en las 15 etiquetas generadas, en la Figura 4.20 se observa el resultado de la clasificación de una sección completa analizada.

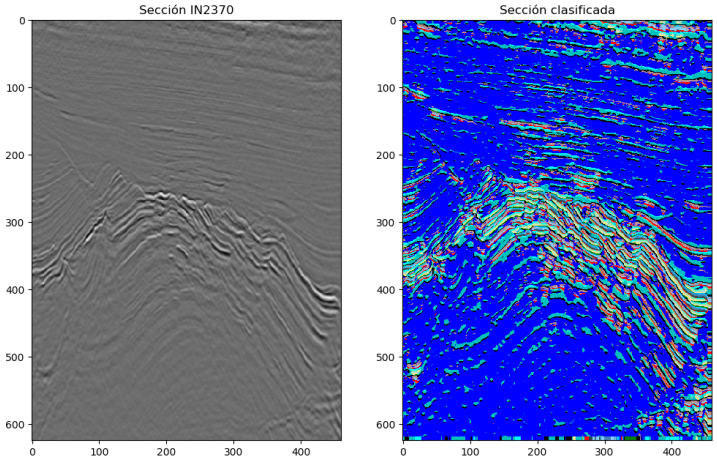


Figura 4.20: Clasificación de la sección sísmica IN2370 después del entrenamiento y división del mapa de neuronas en clases.

En las Figuras 4.18 y 4.20 se muestra la misma sección clasificada pero bajo diferentes estrategias para dividir y crear las etiquetas, no obstante, se observa que en 4.20 las regiones son más homogéneas y prolongadas conservando la topología de los datos sísmicos. En la Figura 4.21 se muestra la etapa tres de la división del mapa de neuronas para clasificar los nuevos de validación.

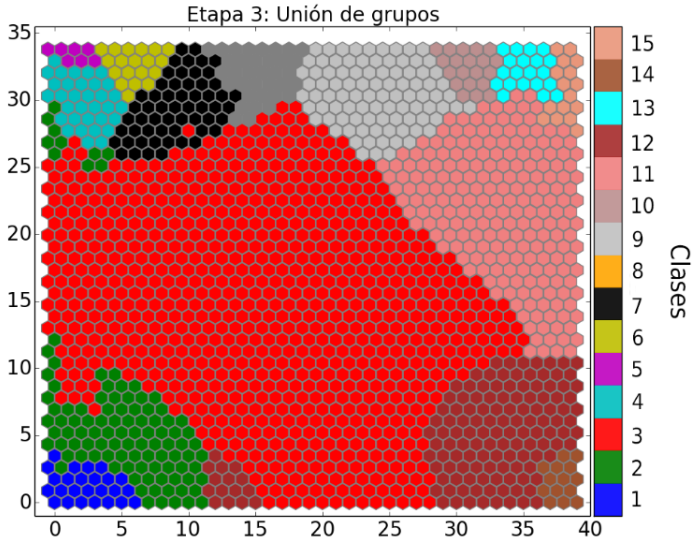


Figura 4.21: Etapa 3 de la división del mapa de neuronas.

En las tres etapas el mapa de neuronas no cambia ya que las divisiones fueron hechas con este número de grupos. Por otra parte, la mayor parte del fondo de la imagen fue mapeada a la clase 1 que tiene poca representación en el mapa 4.21, lo que quiere decir que en esta zona se concentran los patrones que no varían mucho en amplitud y pertenecen a las zonas intermedias entre las capas que se observan en la imagen, de forma contraria la clase 3 que abarca una buena parte del mapa de neuronas se presenta como zonas de cambios en la amplitud (cambios de energía) entre capas como se observa en la Figura 4.20.

Se realizo también un análisis de la señal en potencia para eliminar el factor de la magnitud. En la Figura 4.22 se observa la clasificación de la base de datos utilizando la potencia de la señal para analizarla.

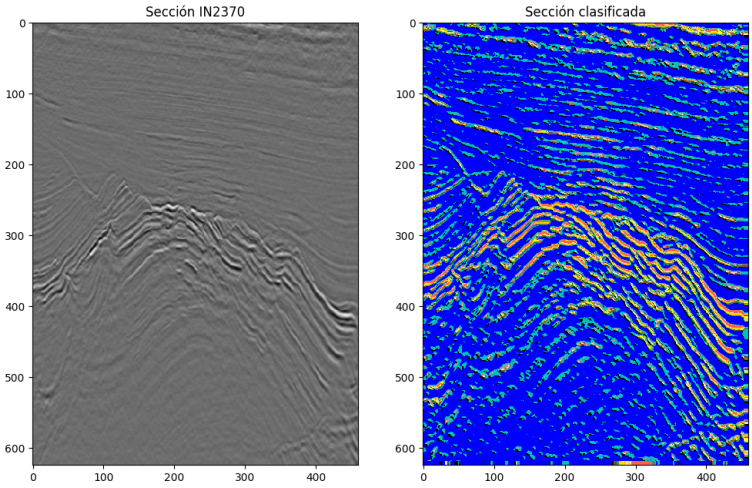


Figura 4.22: Clasificación de las secciones sísmicas utilizando la potencia de la señal.

Como se observa en 4.22 las zonas de cambios mínimos en la amplitud de la señal sísmica se hacen homogéneas al eliminar la magnitud a la señal, por lo que se acentúan mas las zonas con grandes cambios en amplitud (energía), además la estructura de la imagen se muestra más clara y definida. En la Figura 4.23 se observan las clases formadas para clasificar los datos.

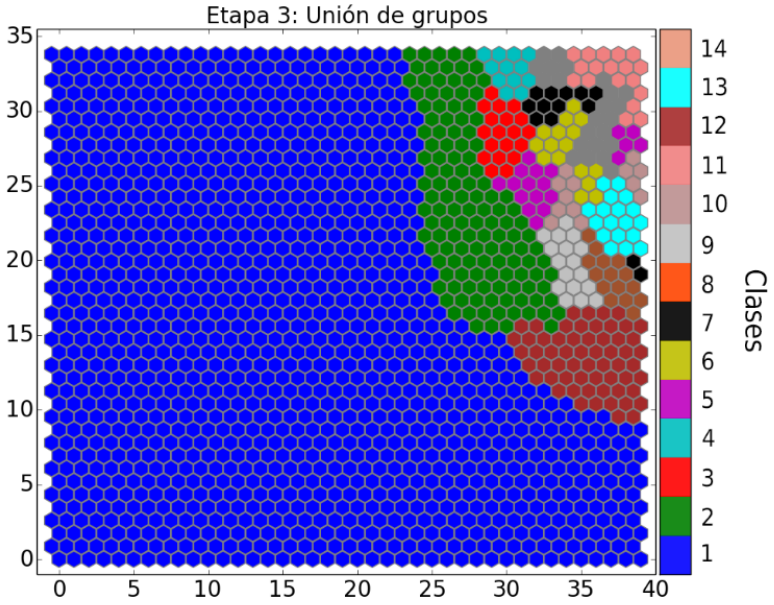


Figura 4.23: Etapa 3 de la división del mapa de neuronas.

En la etapa tres se observa que la mayoría del mapa le pertenece a la clase 1, la cual representa a las zonas con cambios mínimos en la amplitud de la señal. No obstante, las demás clases muestran grandes cambios en la amplitud es donde mayor energía se experimento en la señal la cual se ve reflejada en la topología de los datos sísmicos. En el Anexo B, se muestran todas las etapas de la división del mapa de neuronas y el proceso de clasificación de los datos durante las tres etapas.

Capítulo 5

Conclusión y trabajo a futuro

El problema de clasificar datos cuando no hay información (etiquetas) suficiente para separarlos por clases es una tarea compleja en el área de cómputo. Sin embargo, con la investigación y desarrollo de algoritmos de aprendizaje de máquina se han creado diferentes técnicas computacionales basadas en distintos criterios y metodologías.

En este trabajo de investigación se evaluaron dos metodologías para agrupar la información y con esto generar las clases para clasificar datos: mapas auto-organizados y algoritmos genéticos. Además, se utilizó un conjunto de bases de datos para evaluar el desempeño del modelo en la clasificación de datos de validación, posteriormente se clasificó la base de datos no etiquetada (datos sísmicos) siguiendo una secuencia algo distinta.

En primer lugar, se realizó una investigación del estado del arte de trabajos relacionados a la clasificación de señales sísmicas, así como el uso del algoritmo genético como agrupador de información. Antes de iniciar las metodologías de agrupamiento y clasificación, la información fue pre-procesada para ayudar con la separación de las clases. Las dos técnicas más utilizadas en el trabajo son la normalización y estandarización, ambas fueron aplicadas para separar datos en donde tenían clases que se traslapaban por su dependencia lineal o el cambio de escala, lo cual dificultó el proceso de clasificación. Además dentro del pre-procesamiento se modificó la base de datos sísmica para ser utilizada por las herramientas, ya que los vectores de entrenamiento fueron creados a partir de dividir la base de datos original en pequeñas ventanas con traslape.

Los resultados obtenidos por la metodología de agrupamiento por el algoritmo genético no fueron capaces de generar las clases con las señales de entrada. Se probaron diferentes funciones objetivos y parámetros de configuración y la solución encontrada al problema de optimización planteado no era factible, además de que obtenerla implicaba costos en tiempo y recursos computacionales, ya que para una base de datos de tamaño mediano se obtuvo una imagen sin estructura definida y caótica. Así también la variabilidad en el ajuste de los parámetros y el tiempo entre cada ejecución del experimento no permitieron la realización de varias pruebas para probar diferentes configuraciones, asimismo el error de la función objetivo disminuyó en las primeras generaciones y se estabilizó en un mínimo local después de unas pocas generaciones.

La segunda metodología para agrupar datos que se utilizó fueron los mapas auto-organizados, esta se dividió en tres etapas en donde el mapa de neuronas fue dividido para crear grupos de neuronas que representaron a las clases en las que posteriormente fueron clasificados los datos. Para las primeras dos etapas, se utilizó la distancia euclidiana de los coeficientes de las neuronas para crear a todos los grupos posibles que cumplan con la condición configurada por el usuario, así también se hizo uso de los estadísticos tales como la media, desviación estándar, moda y rango de los grupos previamente creados para unir a los que cumplieran con el porcentaje de similitud seleccionado por el usuario. Por último, se utilizó una medida de evaluación del agrupamiento llamado índice de representación múltiple para unir a los grupos faltantes, esta medida está basada en la distancia entre los grupos y entre los elementos internos de uno mismo para obtener grupos compactos y bien separados.

La metodología de mapas auto-organizados a través de sus tres etapas utilizó medidas de similitud entre grupos para generar clases dentro del mapa de neuronas, es importante señalar que durante las etapas se utilizaron distintos tipos de medidas de similitud, lo que significa que las uniones entre los grupos fueron realizadas siguiendo diferentes enfoques. Además de que en cada etapa siempre existieron validaciones intermedias y finales para asegurar que todas las neuronas fueron correctamente agrupadas a su clase correspondiente.

Las primeras pruebas realizadas fueron con bases de datos etiquetadas para poder medir el desempeño de la metodología basada la división del mapa de neuronas. El desempeño fue medido utilizando las siguientes métricas: error topográfico, error de cuantización, exactitud, precisión, especificidad, sensibilidad y valor F1. Para todas las pruebas realizadas se configuró un tamaño de mapa diferente así como el número de épocas de entrenamiento. Por otra parte, se utilizó una validación cruzada *K-Fold* para validar diferentes mapas entrenados con el conjunto de entrenamiento el cual fue dividido utilizando métodos de Monte Carlo.

Las pruebas con bases de datos etiquetadas están divididas por el tamaño de las bases, para las primeras pruebas las bases de datos no sobrepasaron los 1000 datos, no obstante su dimensionalidad, complejidad y tipo de análisis fueron distintos en cada una. El segundo conjunto de pruebas que se realizaron fueron con bases de datos con más de 1500 datos, empleando mapas más extensos ya que la distribución de los datos en mapas de menor proporción no se logró de manera satisfactoria generando problemas en la clasificación de los datos.

Se obtuvieron buenos resultados con la mayoría de las bases de datos etiquetadas, con un promedio de 90 % en las métricas de evaluación, es decir, la mayoría de los datos de validación fueron clasificados correctamente así como los modelos generados para cada base de datos fueron capaces de identificar los datos correctamente e incorrectamente clasificados. En cuanto a la base de datos de los dígitos (Semeion) se presentó problemas en la clasificación con resultados de 60 % de precisión, sin embargo, el modelo identificó bien a los ejemplos que fueron incorrectamente clasificados por medio de la especificidad.

Una de las propiedades a resaltar de la metodología basada en los mapas autoorganizados fue la conservación de la topología de los datos, ya que para las pruebas de clasificación con las bases de datos control, se observó que las divisiones mostradas en la matriz de distancia unificada coincidían topológicamente con el mapa de neuronas creado después de las tres etapas. Además, en la base de datos tridimensionales quedaron neuronas sin agrupar que representan la distancia de separación entre los cúmulos

de datos de prueba, conservando esa característica topológica de esa base de datos en particular. Por último, esta preservación se observa también al comparar la matriz de coeficientes obtenida con los dígitos y el mapa de neuronas generado, en donde se observa que los números fueron bien identificados espacialmente pero el problema fue la delimitación de las fronteras entre clases, por la superposición de los números durante el entrenamiento y mala ubicación del dígito 9 en el mapa.

Siguiendo con el conjunto de pruebas se evaluó la metodología con la base de datos sísmicos no etiquetada. Para este caso se seleccionó un 20 % de porcentaje de todas las secciones de datos para entrenar y lo demás para clasificar a partir de las clases que fueron creadas. El número de clases fue seleccionado por medio de un análisis visual de los resultados obtenidos con los tres índices de evaluación.

La clasificación de los datos sísmicos que se muestran en los resultados y anexos, se llevaron a cabo conservando la topología de las regiones en las imágenes originales (escala de grises), además de realizar dos tipos de análisis sobre la amplitud de las señales sísmicas permitió identificar zonas más homogéneas y zonas en donde la energía es mayor por grandes cambios en la composición física del subsuelo. No obstante, es necesario de otros tipos de análisis para observar otras características de la datos muestreados.

Si bien la herramienta de mapas auto-organizados esta caracterizada por la conservación de la topología de los datos al momento de mapearlos a la red, es necesario contar con herramientas de visualización de datos en alta dimensión para poder observar estas estructuras complejas. Para esto, se utilizaron técnicas computacionales como proyecciones con tSNE y MDF, así también se desarrolló una herramienta para observar las divisiones por medio de colores hechas en el mapa de neuronas, las cuales muestran estas fronteras entre los grupos de datos con una alta dimensión lo que no resulta en una tarea trivial. Además, se implementó el cálculo del error topográfico para redes con topologías hexagonales, que actualmente no se encuentra desarrollado en la biblioteca de la miniSom y es necesario para medir el desempeño del entrenamiento de los mapas auto-organizados.

El trabajo de investigación ha pasado por varias etapas de desarrollo y experimen-

tación, validando los resultados con bases de datos de prueba y generando nuevas lineas de trabajo a futuro sobre la misma metodología desarrollada, incrementando su operatividad con otro tipo de datos. A continuación se presentan las mejoras a la metodología para trabajos futuros:

- Selección automática de los parámetros configurables de distancia en la creación de grupos en el mapa de neuronas
- Evaluación y selección automática de la cantidad de grupos en los que se dividirá el mapa de neuronas con bases de datos no etiquetadas
- Paralelización de todos los procesos de la metodología
- Entrenamiento intermedio para verificar de nuevo si el número de clases en que se dividió el mapa de neuronas fue el correcto
- Uso de datos etiquetados entre las tres etapas para mejorar la tasa de aprendizaje durante el entrenamiento

Apéndice A

Anexo I: Etapas en base de datos de prueba

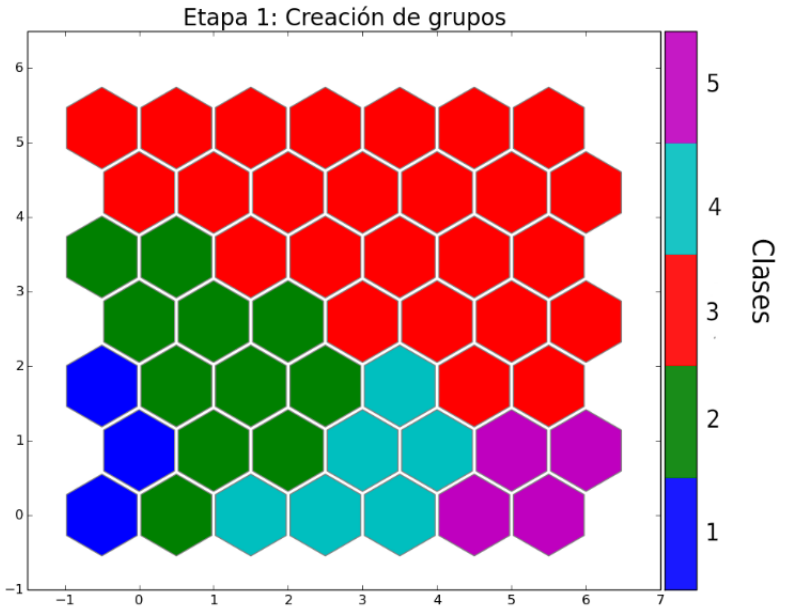


Figura A.1: Etapa 1 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de la ionosfera.

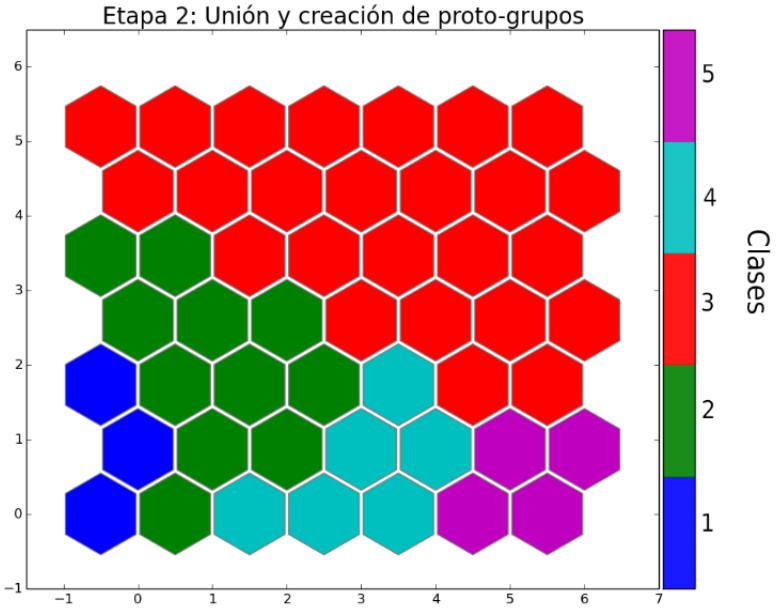


Figura A.2: Etapa 2 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de la ionosfera.

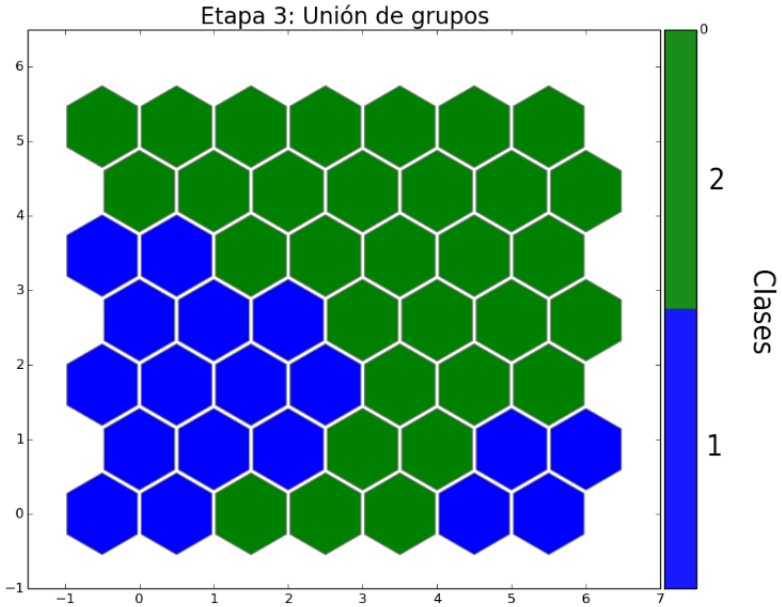


Figura A.3: Etapa 3 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de la ionosfera.

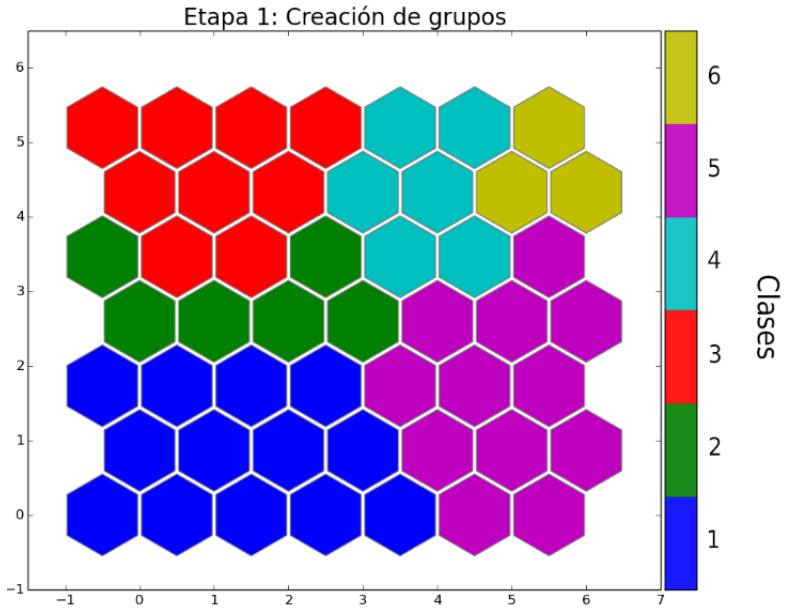


Figura A.4: Etapa 1 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de semillas.

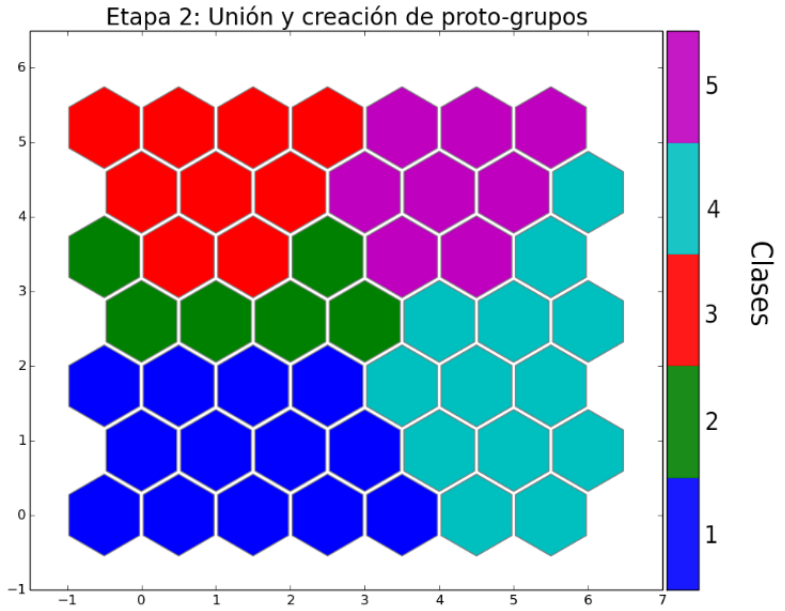


Figura A.5: Etapa 2 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de semillas.

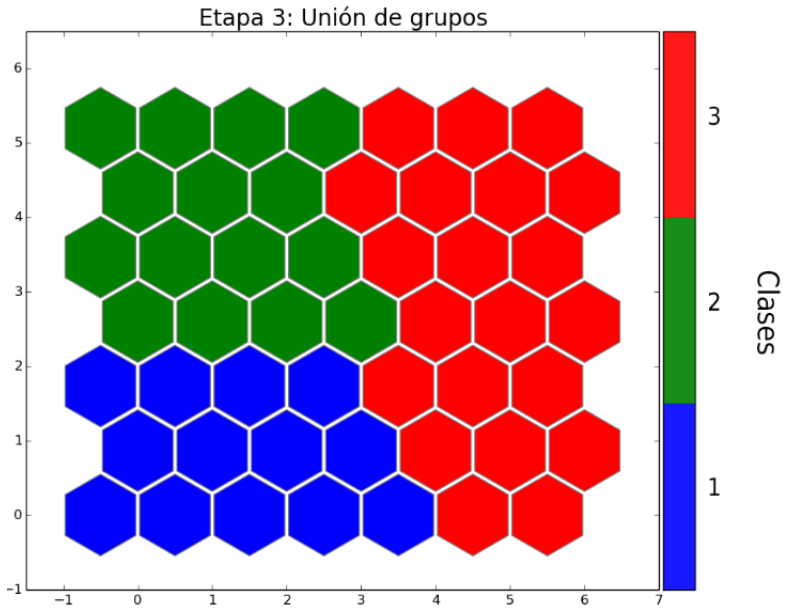


Figura A.6: Etapa 3 del proceso de división del mapa de neuronas en clases para clasificar los datos de la base de datos de semillas.

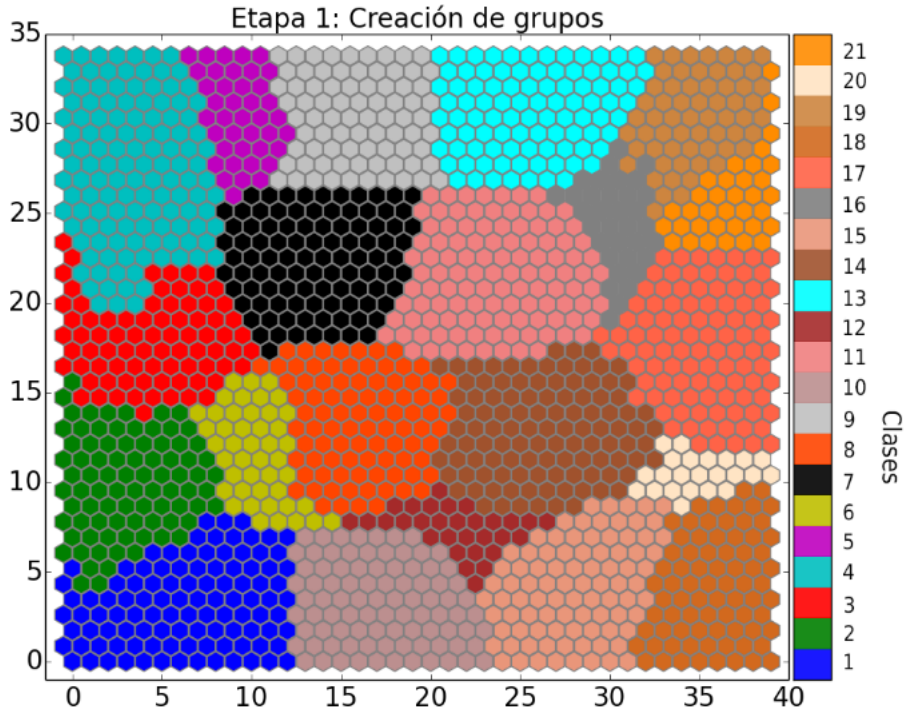


Figura A.7: Etapa 1 del proceso de división del mapa de neuronas en clases para clasificar a los datos bidimensionales.

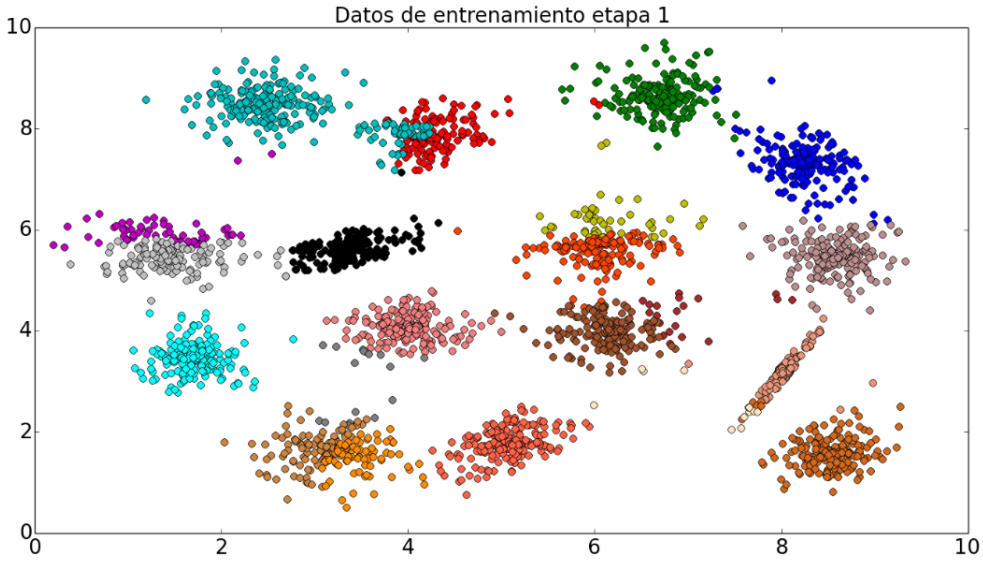


Figura A.8: Datos clasificados con base en el mapa de neuronas.

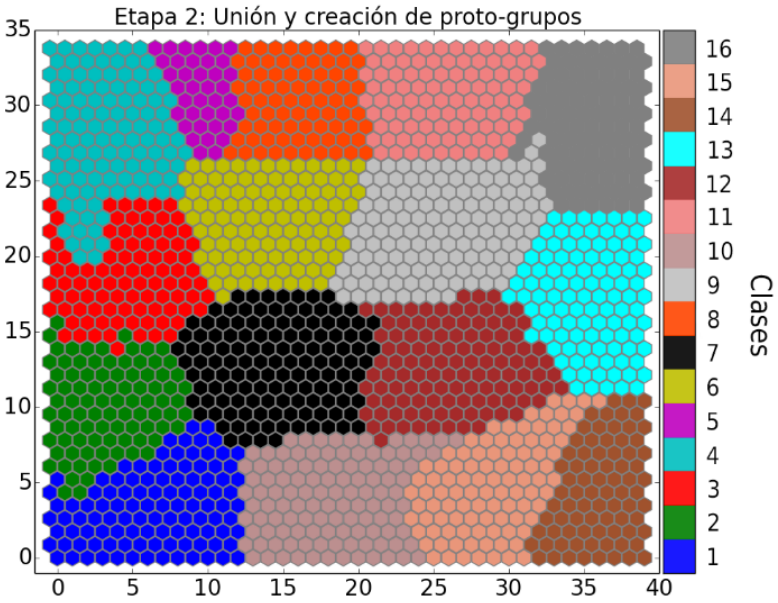


Figura A.9: Etapa 2 del proceso de división del mapa de neuronas en clases para clasificar a los datos bidimensionales.

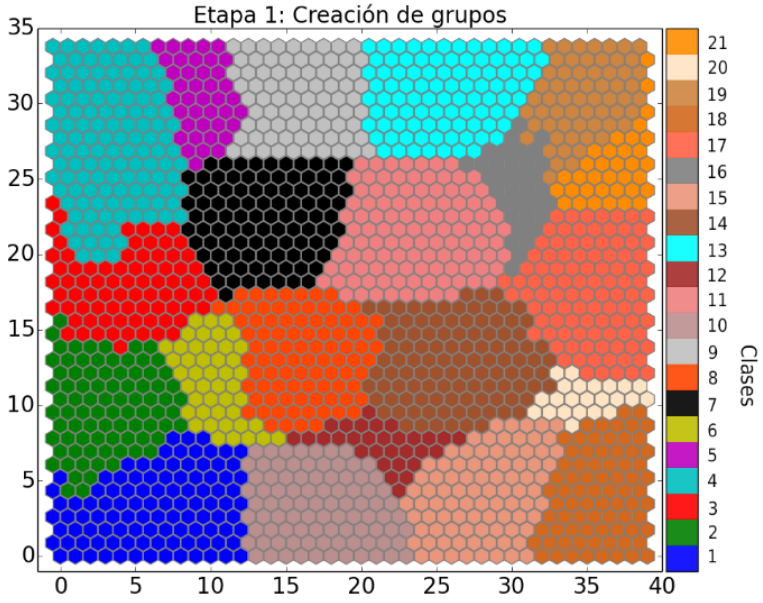


Figura A.10: Etapa 1 del proceso de división del mapa de neuronas en clases para clasificar a los datos bidimensionales.

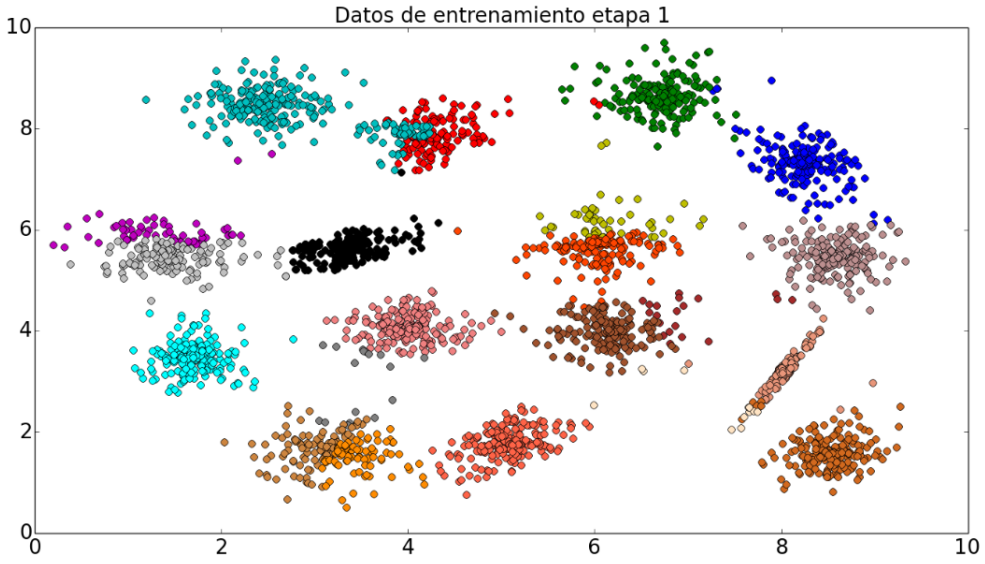


Figura A.11: Datos clasificados con base en el mapa de neuronas.

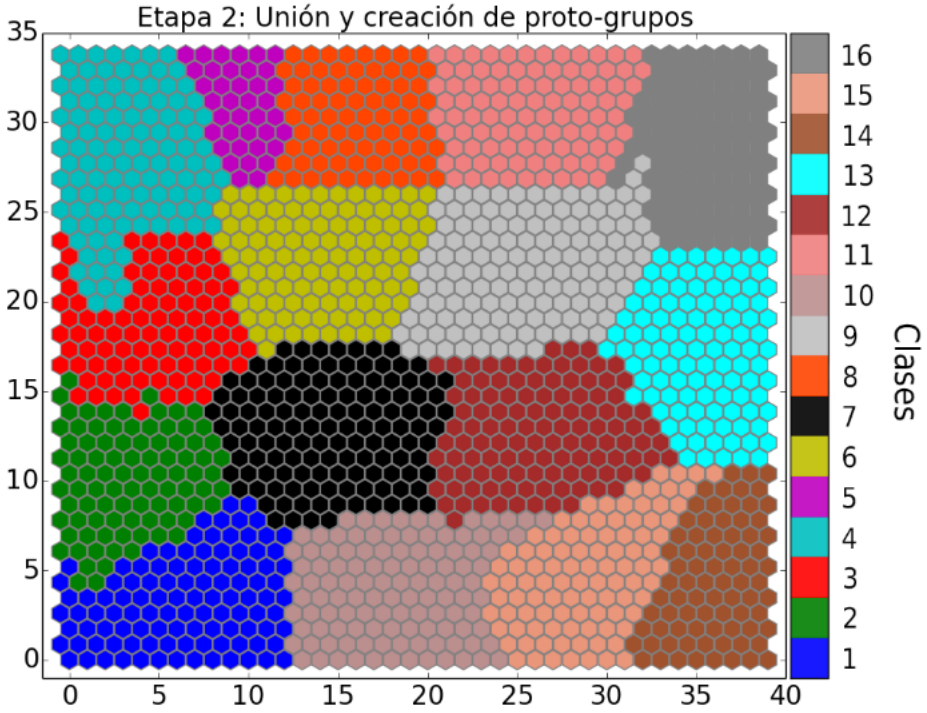


Figura A.12: Etapa 2 del proceso de división del mapa de neuronas en clases para clasificar a los datos bidimensionales.

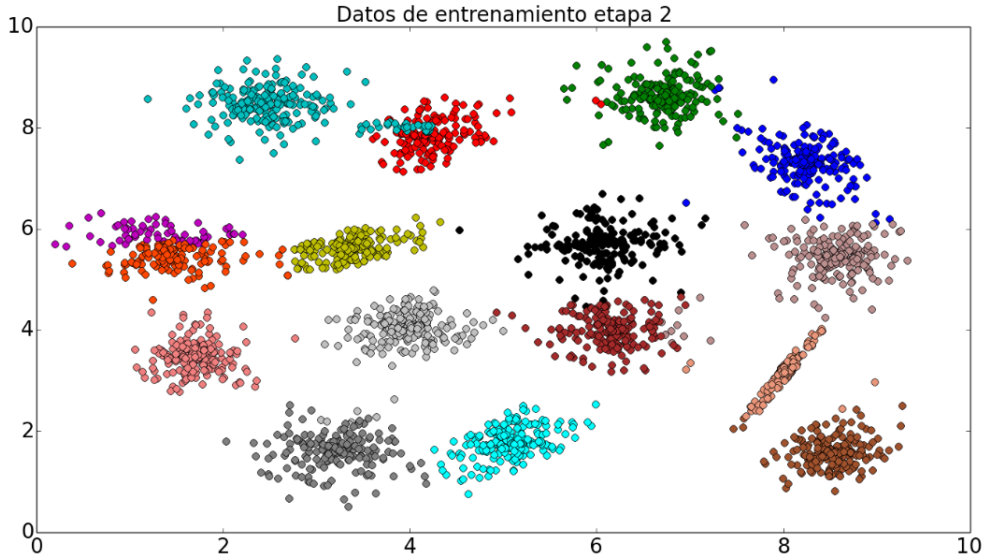


Figura A.13: Datos clasificados con base en el mapa de neuronas.

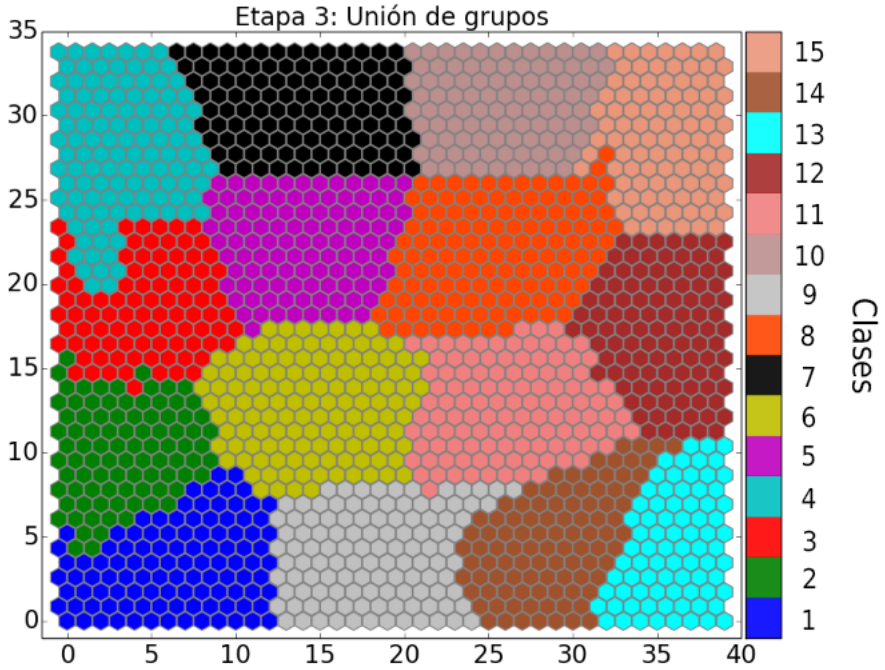


Figura A.14: Etapa 3 del proceso de división del mapa de neuronas en clases para clasificar a los datos bidimensionales.

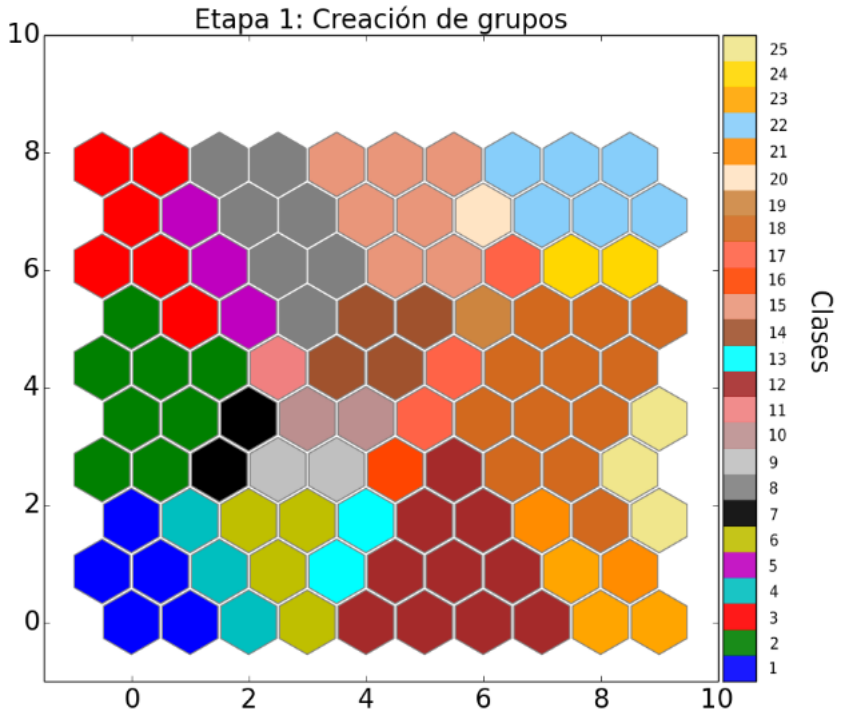


Figura A.15: Etapa 1 del proceso de división del mapa de neuronas en clases para clasificar a los datos tridimensionales.

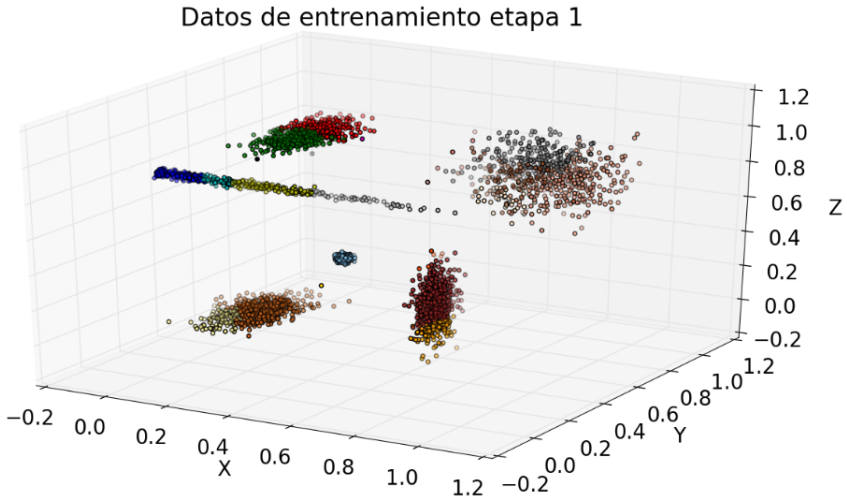


Figura A.16: Datos clasificados con base en el mapa de neuronas.

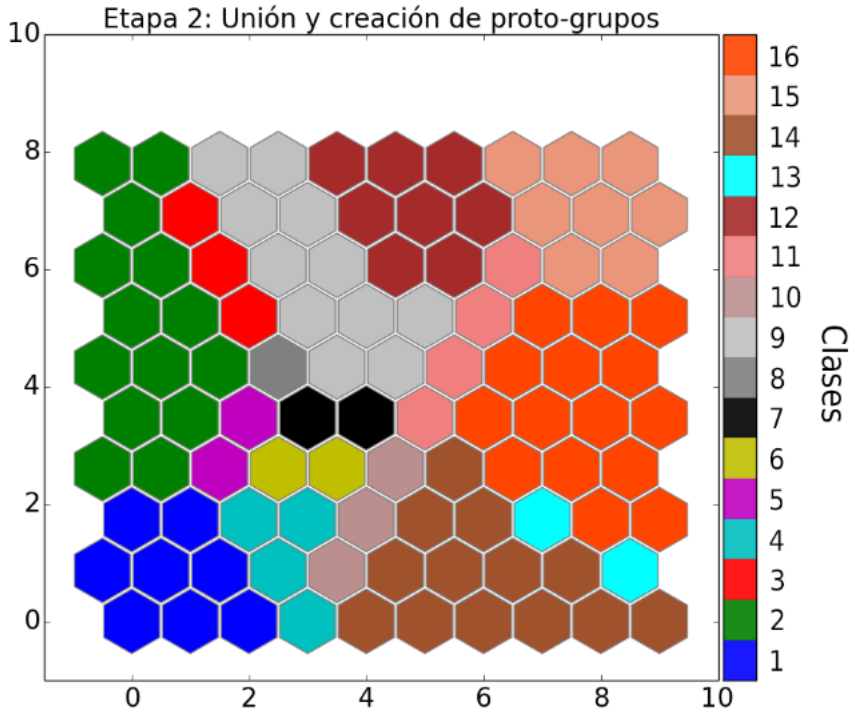


Figura A.17: Etapa 2 del proceso de división del mapa de neuronas en clases para clasificar a los datos tridimensionales.

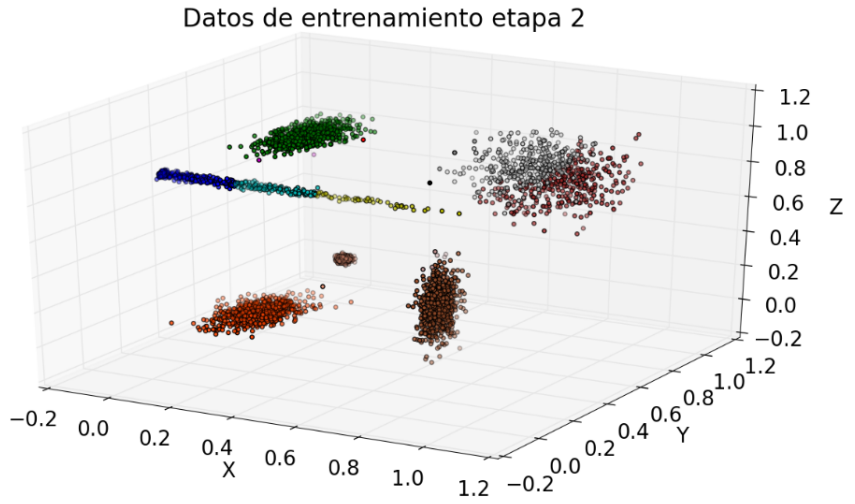


Figura A.18: Datos clasificados con base en el mapa de neuronas.

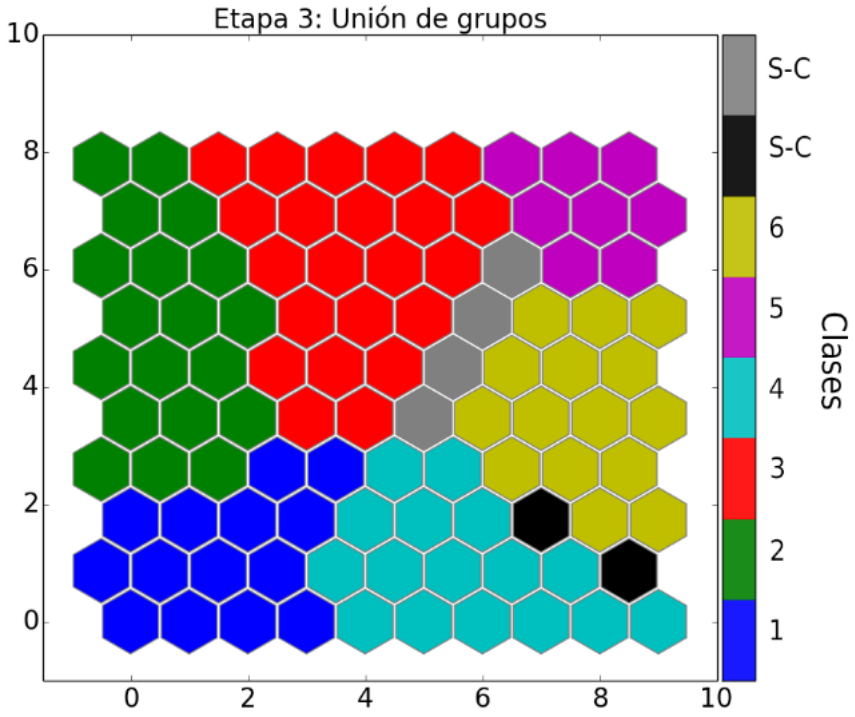


Figura A.19: Etapa 3 del proceso de división del mapa de neuronas en clases para clasificar a los datos tridimensionales.

Apéndice B

Anexo II: Datos sísmicos clasificados

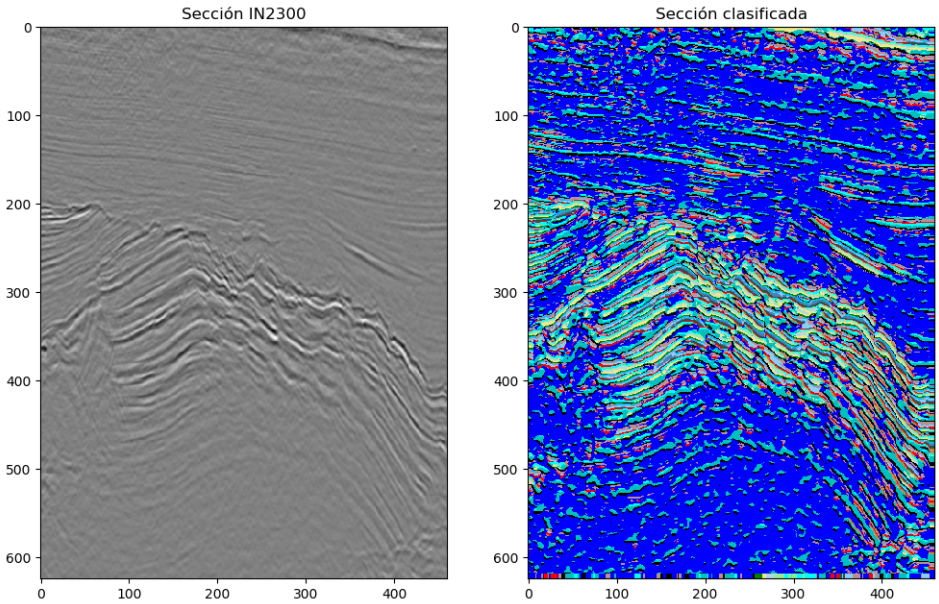


Figura B.1: Clasificación de los datos sísmicos con el modelo basado en el mapa auto-organizado.

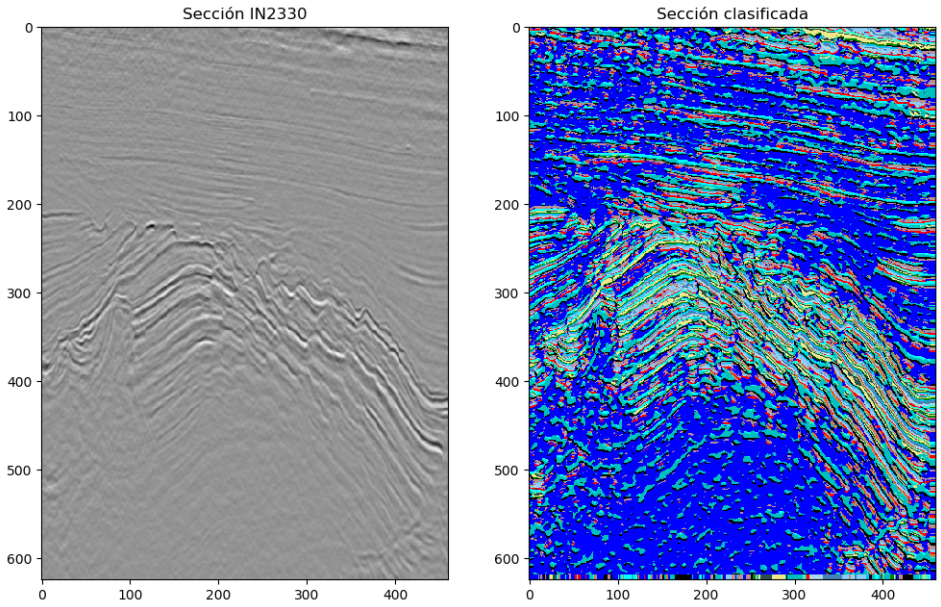


Figura B.2: Clasificación de los datos sísmicos con el modelo basado en el mapa auto-organizado.

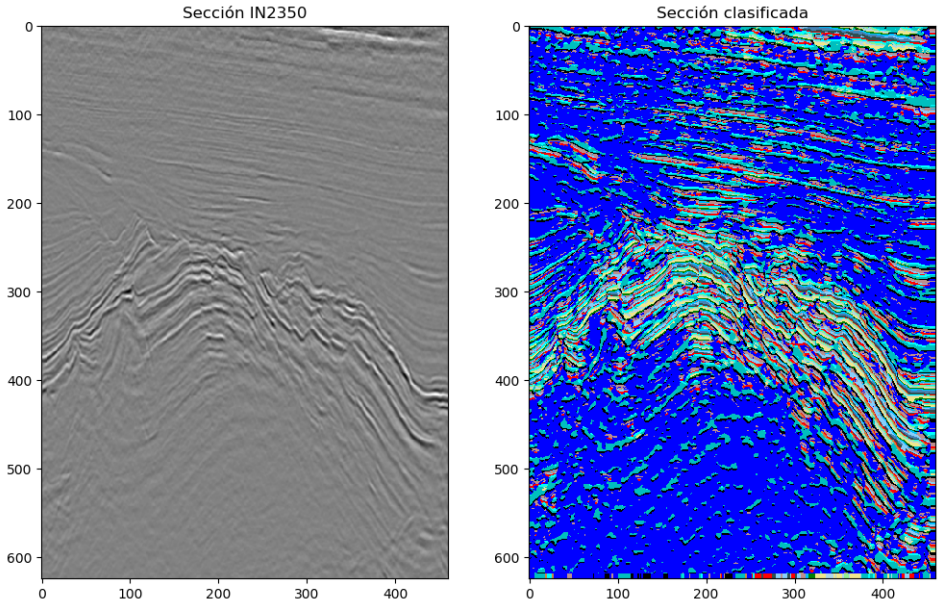


Figura B.3: Clasificación de los datos sísmicos con el modelo basado en el mapa auto-organizado.

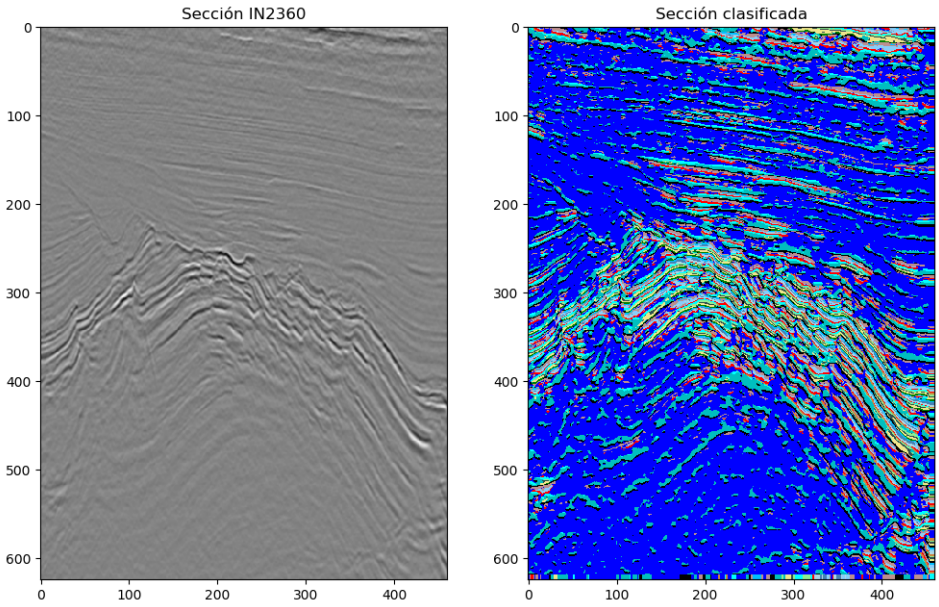


Figura B.4: Clasificación de los datos sísmicos con el modelo basado en el mapa auto-organizado.

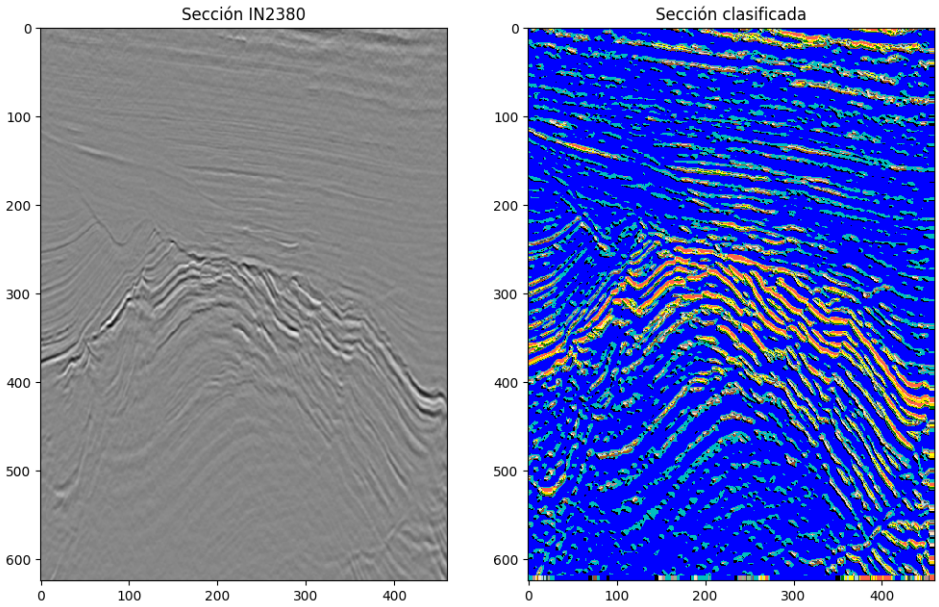


Figura B.5: Clasificación de los datos sísmicos en potencia con el modelo basado en el mapa auto-organizado.

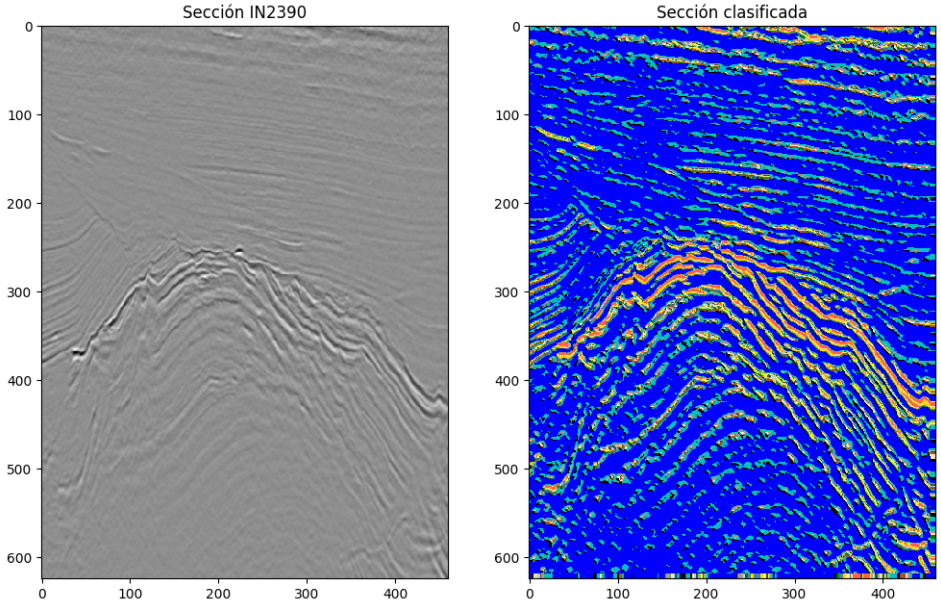


Figura B.6: Clasificación de los datos sísmicos en potencia con el modelo basado en el mapa auto-organizado.

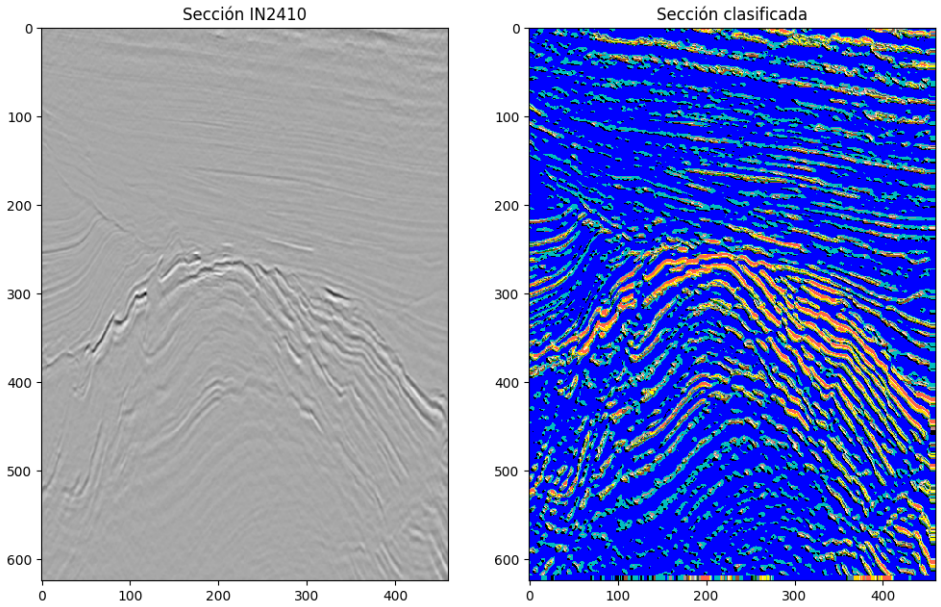


Figura B.7: Clasificación de los datos sísmicos en potencia con el modelo basado en el mapa auto-organizado.

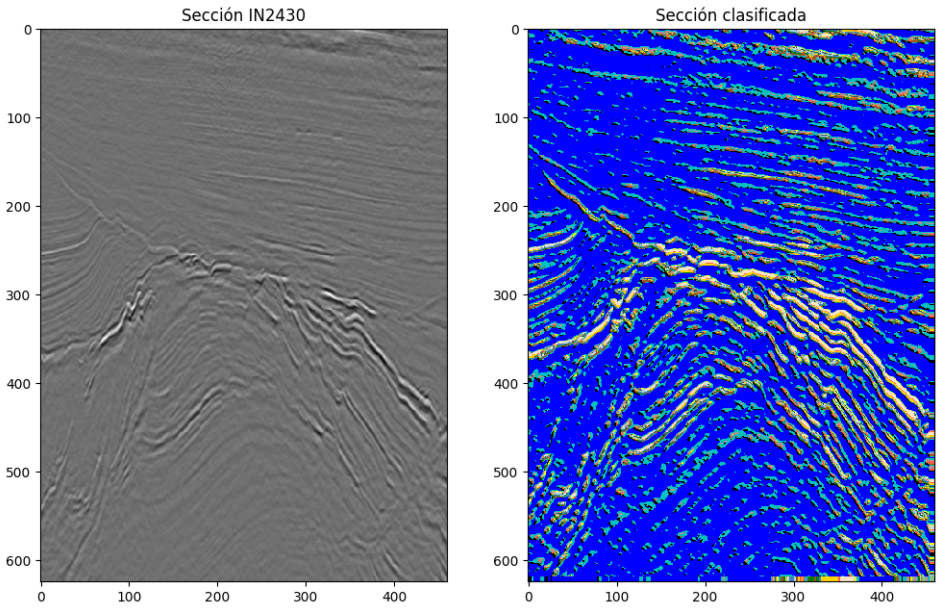


Figura B.8: Clasificación de los datos sísmicos en potencia con el modelo basado en el mapa auto-organizado.

Bibliografía

[1] X. Zhu, "Semi-Supervised Learning Literature Survey Contents," *SciencesNew York*, vol. 10, no. 1530, p. 10, 2008. 2

[2] C. Thurber and K. Aki, "Three-dimensional seismic imaging," *Annual review of earth and planetary sciences. Vol. 15*, vol. 15, pp. 115–139, 11 2003. 2

[3] E. Molino-Minero-Re, E. Rubio-Acosta, H.ítez-Pérez@, J. M. Brandi-Purata, N. I. Pérez-Quezadas, and D. F. García-Nocetti, "A method for classifying pre-stack seismic data based on amplitude-frequency attributes and self-organizing maps," *Geophysical Prospecting*, vol. 66, pp. 673–687, May 2018. 2

[4] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997. 7

[5] E. Alpaydin, *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004. 8, 9, 10

[6] S. Raschka, *Python Machine Learning*. Packt Publishing, 2015. 9

[7] A. B. Tucker, *Computer Science Handbook*. Boca Raton, Fla. : Chapman Hall/CRC, 2004. 9

[8] G. A. Einicke, J. T. Malos, D. C. Reid, and D. W. Hainsworth, "Riccati Equation and EM Algorithm Convergence for Inertial Navigation Alignment," *IEEE Transactions on Signal Processing*, vol. 57, pp. 370–375, Jan. 2009. 10

[9] P. Oliveri, C. Malegori, R. Simonetti, and M. Casale, "The impact of signal pre-processing on the final interpretation of analytical outcomes – a tutorial," *Analytica Chimica Acta*, vol. 1058, Oct 2018. 10

[10] I. Sulis and M. Porcu, "Handling missing data in item response theory. assessing the accuracy of a multiple imputation procedure based on latent classhandling missing data in item response theory. assessing the accuracy of a multiple imputation procedure based on latent class," *Journal of Classification*, vol. 32, pp. 327–359, 07 2017. 11

[11] P. R. Peres-Neto, D. A. Jackson, and K. M. Somers, "How many principal components? stopping rules for determining the number of non-trivial axes revisited," *Computational Statistics Data Analysis*, vol. 49, no. 4, pp. 974–997, 2005. 11

[12] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Ann Arbor, MI: University of Michigan Press, 1975. 12, 14

[13] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* USA: Oxford University Press, Inc., 1996. 13

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989. 13

[15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.).* Berlin, Heidelberg: Springer-Verlag, 1996. 13, 14

[16] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000. 15, 20

[17] J. Silberholz and B. Golden, *The Generalized Traveling Salesman Problem: A New Genetic Algorithm Approach*, pp. 165–181. Boston, MA: Springer US, 2007. 16

[18] T. Kohonen, *Self-Organization and Associative Memory: 3rd Edition*. Berlin, Heidelberg: Springer-Verlag, 1989. 21, 24

[19] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *Trans. Neur. Netw.*, vol. 11, p. 586–600, May 2000. 21, 26

[20] S. M. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *Journal of Industrial Engineering and Management (JIEM)*, vol. 9, no. 2, pp. 374–388, 2016. 22

[21] T. Furukawa, "SOM of SOMs," *Neural Networks*, vol. 22, no. 4, pp. 463–478, 2009. 22

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*. USA: Prentice Hall PTR, 2nd ed., 1998. 23

[23] J. Vesanto, "SOM-Based Data Visualization Methods," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111–126, 1999. 23

[24] G. T.Breard, *Evaluating Self-Organizing Map Quality Measures as Convergence Criteria*. PhD thesis, University of Rhode Island, 2017. 25

[25] S. Wu and T. W. Chow, "Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density," *Pattern Recognition*, vol. 37, no. 2, pp. 175–188, 2004. 26, 29

[26] M. V. Maria Halkidi, "Cluster validity assessment using multi representatives," in *2nd Hellenic conference on AI*, no. April, pp. 237–248, 2002. 26, 27, 29, 30, 31

[27] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, p. 224–227, Feb. 1979. 27

[28] F. K. Iváncsy and Renáta, "A Novel Cluster Validity Index Variance of the Nearest Neighbor Distance," *WSEAS TRANSACTIONS ON COMPUTERS*, vol. 5, no. 3, pp. 477–483, 2006. 27

[29] P. Rousseeuw, "A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov 1987. 28

[30] I. Guyon, K. Bennett, G. Cawley, H. J. Escalante, S. Escalera, Tin Kam Ho, N. Macià, B. Ray, M. Saeed, A. Statnikov, and E. Viegas, "Design of the 2015 chalearn automl challenge," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015. 32

[31] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness correlation," *Mach. Learn. Technol.*, vol. 2, 01 2008. 32, 33

[32] S. Wei and C. P. Soon, "Genetic algorithm-based text clustering technique," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4221 LNCS - I, pp. 779–782, 2006. 36

[33] S. S. Patil and A. S. Bhalchandra, "Pattern recognition using genetic algorithm," *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 310–314, 2017. 37

[34] G. Vettigli, "Minisom: minimalistic and numpy-based implementation of the self organizing map." GitHub.[Online]. Available: https://github.com/JustGlowing/minisom/. 45

[35] A. Ultsch, "Self-Organizing Neural Networks for Visualisation and Classification," no. 1990, 1993. 46

[36] Y. O, *Seismic data processing*. Society of Exploration Geophysicists, 1988. 48

[37] B. V. Dasarathy, "Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, p. 67–71, Jan. 1980. 53

[38] P. Forina, M. et al, "Uci machine learning repository: Wine dataset," 1991. 54

[39] N. Street, W. Wolberg, and O. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," *Proc. Soc. Photo-Opt. Inst. Eng.*, vol. 1993, 01 1999. 56

[40] S. Wing, R. Greenwald, C.-I. Meng, V. Sigillito, and L. Hutton, "Neural networks for automated classification of ionospheric irregularities in hf radar backscattered signals," *Radio Science - RADIO SCI*, vol. 38, pp. 2–1, 08 2003. 56

[41] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," 2018. 57

[42] S. R. C. of Sciences of Communication, "Semeion hanwritten digit data set." 59, 80