



**INDAOE**

**Instituto Nacional de Astrofísica Óptica y  
Electrónica.**

**Reconocimiento fuera de línea de  
palabras manuscritas, basado en un  
Mapa Auto-Organizado**

Por

**Rodolfo Luna Pérez**

Tesis sometida como requisito parcial para obtener  
el grado de

**MAESTRO EN CIENCIAS EN EL ÁREA DE  
CIENCIAS COMPUTACIONALES**  
en el

**Instituto Nacional de Astrofísica Óptica y  
Electrónica.**

Febrero 2011

Tonantzintla, Puebla

Supervisada por:

**Dra. María del Pilar Gómez Gil, INAOE**

©INAOE 2011

Derechos Reservados

El autor otorga al INAOE el permiso de  
reproducir y distribuir copias de esta tesis en su  
totalidad o en partes.



# **Agradecimientos**

A mi asesora Dra. Pilar Gómez Gil por su apoyo y orientación.

A mis sinodales Dr. Aurelio López López, Dr. Luis Villaseñor Pineda y Dr. Manuel Montes y Gómez por sus observaciones y comentarios acertados y sus consejos.

Al INAOE, por todas las facilidades proporcionadas durante mi estancia académica.

A CONACYT por el apoyo económico a través de la beca No. 224412.



# Resumen

En la actualidad grandes volúmenes de información están disponibles en distintos medios electrónicos, sin embargo, todavía existe una gran cantidad de información escrita a mano que se encuentra sólo en papel o en documentos digitalizados (imágenes), e incluso se siguen generando una gran cantidad de este tipo de documentos. Esta situación ha motivado el desarrollo de métodos de reconocimiento de texto manuscrito para recuperar dicha información, ya sea de documentos almacenados (reconocimiento fuera de línea) o reconociendo el texto mientras es generado (reconocimiento en línea). Sin embargo, reconocer texto manuscrito es una tarea muy complicada debido a la variación de los estilos de escritura que existen, siendo aún un problema abierto y sobre el cual se sigue investigando. El problema de reconocer texto manuscrito se ha atacado de distintas formas: reconociendo caracteres, reconociendo palabras e incluso reconociendo líneas de texto completas.

En este documento se propone un método novedoso para el reconocimiento fuera de línea de palabras manuscritas, que aprovecha las ventajas que ofrecen las redes neuronales recurrentes para la clasificación de datos que dependen del tiempo, y la auto-organización de las redes neuronales SOM para agrupar segmentos de palabra similares y utilizar esta información de similitud para lograr una mejor clasificación. El método propuesto utiliza una red recurrente simple, también conocida como “red Elman”, la cual es alimentada por un vector de características obtenido de la aplicación de una función de distribución de probabilidad sobre los resultados de activación de un mapa SOM. El método que se presenta fue probado utilizando un sub-conjunto de la base de datos IAM a través de varios experimentos, encontrándose que la combinación de redes SOM, medidas de similitud y redes neuronales recurrentes mejoran sustancialmente la clasificación de palabras manuscritas. El

clasificador construido obtuvo un promedio de 78.2% de exactitud en el reconocimiento de 10 palabras distintas, en comparación al 66.2% obtenido por un reconocedor basado únicamente en una combinación de redes SOM y perceptrones de varios niveles, y al 32.1% obtenido por un clasificador basado únicamente en perceptrones.

# Abstract

Nowadays there is a huge amount of information available in many different electronic devices; however, there is still a lot of information that only can be found in non-digital handwritten documents or handwritten documents digitalized as images; and even diverse activities produces yet handwritten documents. This has motivated the development of handwritten text recognition methods to retrieve such information, whether stored documents (off-line recognition) or by recognizing the text as it is generated (online recognition). However, handwriting recognition is a challenging task due to the variation of writing existing styles and it is still an open problem that is in research. The handwriting recognition problem has been attacked in several ways: recognizing characters, recognizing words and even recognizing whole lines of text.

This document proposes a novel method for off-line handwritten word recognition, which takes advantage of recurrent neural networks for classification of time-dependent data, and self-organization of the SOM neural networks in clustering similar word segments and use this information of similarity to improve the process of classification. The proposed method uses a simple recurrent network, also known as Elman network, which is fed by a vector of features obtained from the application of a probability distribution function on the outcome of activation of a SOM map. The method presented was tested using a subset of the IAM database through several experiments, the results obtained showed that the combination of a SOM network, similarity measures and recurrent neural networks significantly improve the classification of handwritten words. The classifier trained had an average of 78.2% of accuracy in recognition of 10 classes of words, compared to 66.2% obtained by a recognizer based solely on a combination of SOM networks and multi-layered perceptron, and 32.1% obtained by a multi-layer perceptron classifier.

# Índice General

<b>Resumen</b> .....	<b>I</b>
<b>Abstract</b> .....	<b>III</b>
<b>Índice de Figuras</b> .....	<b>IX</b>
<b>Índice de Tablas</b> .....	<b>XI</b>
<b>1. Introducción</b> .....	<b>1</b>
1.1 Descripción del Problema .....	1
1.2 Objetivos .....	3
1.2.1 Objetivo General .....	3
1.2.2 Objetivos Específicos.....	3
1.3 Metodología de Investigación.....	3
1.4 Aportaciones .....	5
1.5 Estructura del documento.....	6
<b>2. Marco Teórico</b> .....	<b>7</b>
2.1 Reconocimiento de Texto Manuscrito .....	7
2.2 Preprocesamiento de imágenes con texto manuscrito.....	9
2.2.1 Binarización de imágenes .....	10
2.2.2 Corrección de la inclinación.....	10
2.3 Segmentación .....	13
2.4 Clasificación de secuencias temporales .....	14

2.4.1	Técnicas para la clasificación de secuencias temporales.....	15
2.5	Redes Neuronales Artificiales .....	16
2.5.1	Red alimentada hacia adelante .....	18
2.5.2	Redes neuronales recurrentes .....	18
2.6	Mapas Auto-Organizados .....	19
<b>3.</b>	<b>Trabajo relacionado .....</b>	<b>23</b>
3.1	Trabajos que utilizan modelos ocultos de Markov .....	23
3.2	Trabajos relacionados basados en redes neuronales artificiales .....	27
<b>4.</b>	<b>Método Propuesto.....</b>	<b>35</b>
4.1	Descripción General .....	35
4.1.1	Preprocesamiento.....	36
4.1.2	Segmentación de las imágenes .....	37
4.1.3	Agrupamiento y extracción de características .....	38
4.1.4	Clasificación de palabras .....	42
4.1.5	Ejemplo del reconocimiento de palabras .....	45
<b>5.</b>	<b>Experimentos y Resultados.....</b>	<b>49</b>
5.1	Evaluación de los resultados .....	49
5.1.1	Error de reconocimiento .....	49
5.1.2	Exactitud de palabra ( <i>word accuracy</i> ).....	50
5.2	Descripción de la base de datos .....	50
5.3	Experimentos realizados.....	51
5.3.1	Experimentos con segmentación manual .....	52
5.3.2	Experimentos con segmentación automática.....	54
5.3.3	Experimentos adicionales .....	58
<b>6.</b>	<b>Conclusiones y trabajo futuro .....</b>	<b>61</b>



6.1	Conclusiones .....	61
6.2	Trabajo futuro.....	62
	<b>Publicaciones derivadas del Trabajo de tesis .....</b>	<b>64</b>
	<b>Referencias.....</b>	<b>65</b>
	<b>Apéndice A.....</b>	<b>69</b>
	<b>Apéndice B.....</b>	<b>73</b>
	<b>Apéndice C.....</b>	<b>77</b>
	<b>Apéndice D.....</b>	<b>79</b>
	<b>Apéndice E.....</b>	<b>81</b>



# Índice de Figuras

Figura 2.1 Etapas del proceso de reconocimiento de texto manuscrito propuesto por Gómez-Gil y colaboradores (2007).....	9
Figura 2.2 Ejemplo de binarización de una imagen. (a) imagen tomada de la base de datos IAM-DB (Marti & Bunke, 1999). (b) imagen binarizada. ....	10
Figura 2.3 Ejemplo de una palabra con inclinación de trazo $\alpha$ (imagen tomada de la base de datos IAM DB (Marti & Bunke, 1999)).....	11
Figura 2.4 Ejemplo de corrección de inclinación (Toselli, 2004).....	12
Figura 2.5 Direcciones utilizadas para la representación de imágenes con código de cadena (Ding et al., 2004). ....	13
Figura 2.6 Ejemplo de la corrección de inclinación (Ding et al., 2004). ....	13
Figura 2.7 Modelo de una neurona artificial (Sierra Araujo, 2006).....	17
Figura 2.8 Red neuronal genérica (Sierra Araujo, 2006).....	17
Figura 2.9 Arquitectura de la Red SOM (Kohonen, 2001). ....	21
Figura 3.1 Modelo oculto de Markov de 7 estados propuesto por El-Yacoubi y colaboradores (1999), para el modelado de un carácter.....	24
Figura 3.2 Modelo de Markov de 3 estados que modela la letra manuscrita "b". Los números sobre los vértices indican la probabilidad de transición de un estado a otro (Toselli, 2004). ....	25
Figura 3.3 Autómata estocástico de estados finitos que modela la palabra "mil" y "Mil" (Toselli, 2004).....	26
Figura 3.4 Autómata estocástico de estados finitos con transiciones reemplazadas por los modelos de cada carácter (Toselli, 2004). ....	26
Figura 3.5 Modelo propuesto por Graves y colaboradores (2009) .....	28
Figura 3.6 Proceso completo para el reconocimiento de una palabra (Murillo Gil, 2007). ....	29
Figura 3.7 Líneas base en una palabra. ....	30

Figura 3.8 Esquema general del método propuesto por Cote y colaboradores (1998). .....	32
Figura 3.9 Sistema para el reconocimiento de palabras propuesto por Namane y colaboradores (2005). .....	33
Figura 4.1 Método propuesto para la clasificación de una palabra. ....	37
Figura 4.2 Segmentación automática propuesta. La palabra se muestra en gris para resaltar los pixeles negros inferiores de cada columna. Las líneas verticales azules marcan los puntos de segmentación. ....	38
Figura 4.3 Mapa SOM generado a partir de caracteres segmentados manualmente. ...	40
Figura 4.4 Arquitectura de una red recurrente simple (Elman, 1990). .....	43
Figura 4.5 a) Imágen original de la palabra “But” obtenida de la base de datos IAM DB. b) Imágen resultante después de aplicar el algoritmo de preprocesamiento a la imagen original. ....	45
Figura 4.6 Mapa SOM entrenado a partir de segmentos de palabra generados automáticamente. ....	45
Figura 4.7 Activación de las neuronas de salida, para los 5 segmentos presentados que forman la palabra "But", los títulos de las series corresponde a la palabra representada por cada neurona en la capa de salida. ....	46
Figura 4.8 Activación de las neuronas de salida al presentarle a la red los vectores Ft correspondientes las palabras: a) and, b) as, c) at .....	47
Figura 5.1 Exactitud de palabra promedio obtenido con la SRN para el conjunto de entrenamiento (segmentación manual). ....	55
Figura 5.2 Exactitud de palabra promedio obtenido con la SRN para el conjunto de prueba (segmentación manual). ....	55
Figura 5.3 Exactitud de palabra promedio obtenido con el conjunto de entrenamiento para SRN's con distinto número de nodos en la capa culta (segmentación automática). .....	56
Figura 5.4 Exactitud de palabra obtenido con el conjunto de prueba para SRN's con distinto número de nodos en la capa culta (segmentación automática). ....	57
Figura 5.5 Exactitud de palabra promedio para $k=1...10$ .....	59

# Índice de Tablas

Tabla 3.1. Porcentaje de reconocimiento obtenido en el trabajo de El-Yacoubi y colaboradores (1999).....	25
Tabla 3.2 Porcentaje de reconocimiento tomando N palabras candidatas (Côté et al., 1998). .....	32
Tabla 4.1 Vectores de características generados a partir de la red SOM de acuerdo al método propuesto. ....	46
Tabla 5.1 Vocabulario utilizado en los experimentos. ....	51
Tabla 5.2 Error de Reconocimiento para datos segmentados manualmente. ....	53
Tabla 5.3 Exactitud de palabra para datos segmentados manualmente. ....	53
Tabla 5.4 Error de reconocimiento para 10 palabras distintas segmentadas automáticamente. ....	56
Tabla 5.5 Exactitud de palabra obtenido con segmentación automática. ....	56
Tabla 5.6 Error de reconocimiento para 10 palabras distintas segmentadas automáticamente y agregando al vector de características el tamaño de la secuencia. ....	58
Tabla 5.7 Exactitud de palabra obtenido agregando el tamaño de la secuencia al vector de características. ....	58
Tabla 5.8 Exactitud de palabra promedio y porcentaje de error promedio para el reconocimiento de 10 palabras con $k=1$ y segmentación automática. ....	59
Tabla 5.9 Porcentaje de error promedio y exactitud de palabra promedio para el reconocimiento de 10 palabras utilizando el método propuesto con $k=5$ , sin función de probabilidad y con segmentación automática. ....	60
Tabla E.1 Porcentaje de error promedio y exactitud de palabra con sus respectivas desviaciones estándar obtenidos variando el número de nodos en la capa oculta de la red FF para el reconocimiento de palabras. Segmentación manual. ....	81

Tabla E.2 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red SOM-FF con $k=5$ y segmentación manual. ....	85
Tabla E.3 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con $k=5$ y segmentación manual. ....	88
Tabla E.4 Porcentaje de error promedio y exactitud de palabra con sus respectivas desviaciones estándar obtenidos variando el número de nodos en la capa oculta de la red FF para el reconocimiento de palabras. Segmentación automática. ....	91
Tabla E.5 Exactitud de palabra y porcentaje de error de reconocimiento promedio la red SOM-FF con $k=5$ y segmentación automática. ....	92
Tabla E.6 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con $k=5$ y segmentación automática. ....	93
Tabla E.7 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red FF con segmentación automática y utilizando el número de elementos que componen a la secuencia como característica. ....	94
Tabla E.8 Porcentaje Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red SOM-FF con $k=5$ , segmentación automática y utilizando el número de elementos que componen a la secuencia como característica. ....	95
Tabla E.9 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con $k=5$ , segmentación automática y utilizando el número de elementos que componen a la secuencia como característica. ....	96
Tabla E.10 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con $k=1$ , sin utilizar la función de probabilidad y con segmentación automática. ....	96
Tabla E.11 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con $k=5$ , sin utilizar la función de probabilidad y con segmentación automática. ....	97

# Capítulo 1

## Introducción

### 1.1 Descripción del Problema

El reconocimiento de texto manuscrito fuera de línea es una parte importante para la tarea de recuperación de información que tiene varias aplicaciones como: reconocimiento de códigos postales, reconocimiento de cantidades escritas en cheques, obtención de la información de encuestas y formularios, reconocimiento de textos antiguas sin la necesidad de un experto, identificación o autenticación de personas y otras. Además, muchas tareas actuales siguen produciendo documentos con texto manuscrito. Sin embargo, el reconocimiento de texto manuscrito es una tarea complicada debido a la variedad en los estilos de escritura, dicha variedad hace complejas varias tareas asociadas con el reconocimiento de texto manuscrito, como son la segmentación, reconocimiento de caracteres y reconocimiento de palabras, entre otros. Por otro lado, el reconocimiento de texto manuscrito a nivel de caracteres presenta muchos problemas, por lo que, el construir palabras a partir de los caracteres reconocidos ocasiona que el reconocimiento a nivel de palabras tenga un error muy alto, debido a esto, se ha buscado reconocer palabras completas, en lugar de caracteres aislados, para mejorar el reconocimiento.

En los últimos años se han desarrollado varios métodos para atacar el problema del reconocimiento de palabras manuscritas, la mayoría están basados en los Modelos Ocultos de Markov (HMM) y Redes Neuronales Artificiales. Los trabajos que emplean HMM están basados en los métodos desarrollados para el reconocimiento del habla argumentado que son problemas similares. Por otro lado, el uso de redes

neuronales también ha mostrado tener buenos resultados, además uno de los últimos trabajos desarrollados para el reconocimiento de texto manuscrito, basado en redes neuronales, mostró un mejor desempeño que los HMM bajo el argumento de que una red neuronal recurrente permite utilizar más información del contexto que un HMM que sólo toma en cuenta un estado anterior.

Uno de las grandes dificultades que presenta el reconocimiento de texto manuscrito es la similitud que pueden tener los caracteres o los segmentos en los que se dividen las imágenes de texto, esto hace que para el clasificador sea difícil distinguir que carácter es o a que carácter corresponde dicho segmento. Por lo tanto, se debe buscar la forma de atacar esta dificultad.

Los Mapas Auto-organizados (SOM) son redes neuronales que se entrenan con datos no etiquetados generando prototipos, que representan a grupos, y están ordenados en un arreglo bidimensional. El orden está basado en la similitud entre los prototipos generados, esta similitud puede ser utilizada para proporcionar información importante al clasificador. Las redes SOM también han sido utilizadas en el análisis del habla.

El trabajo que se expone en este documento se enfoca en probar un método que permita utilizar la información de semejanza entre las características que definen a los trazos manuscritos, para ser usada en el proceso de reconocimiento de palabras manuscritas. Esta semejanza se obtiene al utilizar una red neuronal tipo SOM en la clasificación de segmentos de palabra, de la cual se obtenga información probabilística que permita a su vez reconocer palabras a través de un reconocedor basado en redes neuronales recurrentes. El reconocimiento de palabras se apoya en un léxico previamente definido.

Esta investigación forma parte del proyecto PRISCUS (Cuevas & Gómez-Gil, 2008), desarrollado en el laboratorio de redes neuronales y reconocimiento de patrones de la coordinación de computación del INAOE, cuyo objetivo es construir un software inteligente e integral para transcripción de textos manuscritos y antiguos.



## **1.2 Objetivos**

### **1.2.1 Objetivo General**

El objetivo de esta investigación es la construcción de un método para reconocimiento de palabras manuscritas, independiente del escritor y con vocabulario definido, que muestre que el uso de información de segmentos procesados por una red neuronal SOM puede mejorar el reconocimiento a nivel de palabra. Se entiende por segmento a una porción de la palabra, obtenido a través de un proceso automático, que puede o no coincidir con un carácter.

### **1.2.2 Objetivos Específicos**

Para alcanzar el objetivo planteado, se definieron los siguientes objetivos específicos:

- Definir una función de distribución que permita representar la probabilidad de pertenencia de cada segmento de una palabra a  $k$  posibles grupos vecinos de una red neuronal SOM, previamente entrenada para clasificación no supervisada de segmentos de palabra. El corte de cada segmento se lleva a cabo utilizando el método descrito en la sección 4.1.2.
- Construir un clasificador que aproveche las cualidades de las redes neuronales y permita el reconocimiento de secuencias de datos. Dicho clasificador recibirá como entrada secuencias de vectores, que representan a los segmentos de las palabras, y las clasifique utilizando la información contextual inherente en estas secuencias.

## **1.3 Metodología de Investigación**

Para conseguir los objetivos planteados se realizó lo siguiente:

- Selección de técnicas de preprocesamiento.

Se determinó a través de un análisis de trabajos relacionados que el mejor preprocesamiento para este tipo de problema consiste en binarización, eliminación de ruido y corrección de la inclinación de trazos. Para llevar a cabo la binarización se eligió el operador umbral debido, a que es un método sencillo y comúnmente utilizado. Para eliminar el ruido de las imágenes se utilizó el filtro mediana.

Para seleccionar el método para la corrección de inclinación se implementaron y analizaron los métodos propuestos por Toselli (2004) y Ding y colaboradores (2004), eligiendo este último debido a que mostró mejores resultados.

- Selección de datos de prueba.

Para realizar los experimentos se eligió la base de datos IAM-DB (Marti & Bunke, 1999); esta base es de uso libre para la investigación y fue utilizada en uno de los trabajos más recientes en reconocimiento de palabras (Graves et al., 2009).

Para la evaluación de los resultados obtenidos se calculó el porcentaje de reconocimiento y la métrica “exactitud de palabra” (*word accuracy*) los cuales se definen en la sección 5.1.

- Análisis de modelos neuronales.

Se analizaron distintos modelos de redes neuronales para elegir aquél que pudiera ser utilizado para objetos descritos como secuencias temporales. Como resultado de este análisis se eligió una red neuronal recurrente simple propuesta por Elman (1990). El algoritmo de entrenamiento utilizado fue retro-propagación a través del tiempo, basado en el gradiente descendiente y con factor de aprendizaje. El entrenamiento se hizo de manera incremental, de tal forma que aprendiera secuencias independientes correspondientes a cada una de las palabras.

- Diseño de la función de distribución.

Se diseñó una función de distribución basada en la función vecindad utilizada en el algoritmo de entrenamiento de la red SOM. La función propuesta permite obtener la probabilidad de que un segmento de la imagen de una palabra

pertenezca a los grupos de cierta región del mapa de salida de la red SOM; dicha región está definida por  $k$  neuronas con mayor activación. La función es descrita a detalle en la sección 4.1.3.

- Diseño de la red neuronal.

La red neuronal recurrente simple utilizada consta de 3 capas: capa de entrada, capa oculta y capa de salida. La capa de entrada tiene  $3k$  nodos de entrada en donde  $k$  es el número de neuronas de la capa de salida del mapa SOM tomados en cuenta para el cálculo de la función de distribución. La capa de salida cuenta con un nodo por cada palabra que se quiere reconocer (ver sección 4.1.4).

- Experimentación y aporte de modelos.

Se diseñaron experimentos que permitieron observar la mejora del reconocimiento al utilizar la red neuronal SOM como fase intermedia en el reconocimiento de palabras.

## 1.4 Aportaciones

En este trabajo se realizaron las siguientes aportaciones.

- 1) Se propuso una función para obtener información probabilística, de pertenencia de un segmento a un grupo determinado a partir de un Mapa Auto-Organizado (SOM). La métrica está inspirada en la función de vecindad utilizada en la fase de entrenamiento de la red.
- 2) Se propuso un método de segmentación de palabras que generara segmentos con las características adecuadas para el método propuesto.
- 3) Se propuso la integración la función propuesta, la salida de la red SOM y una red neuronal recurrente cuya arquitectura y entrenamiento permite la clasificación de secuencias temporales, para el reconocimiento de texto manuscrito a nivel de palabras.

## **1.5 Estructura del documento**

Este documento está organizado en 6 capítulos. En el capítulo 2 se presentan los conceptos básicos relacionados con este trabajo de investigación. En el capítulo 3 se describen algunos de los trabajos más relevantes relacionados con el reconocimiento de palabras manuscritas. En el capítulo 4 se describe el método propuesto y las distintas etapas que lo componen; asimismo, se describe el entrenamiento de las redes neuronales que están involucradas en el método. En el capítulo 5 se describen los experimentos que se diseñaron para probar el método propuesto y se exponen los resultados obtenidos. En el capítulo 6 se presentan las conclusiones obtenidas de la investigación y el posible trabajo futuro.

# Capítulo 2

## Marco Teórico

En este capítulo se presentan algunos conceptos fundamentales que ayudan al entendimiento del trabajo de investigación y al resto de los capítulos contenidos en este documento.

### 2.1 Reconocimiento de Texto Manuscrito

El reconocimiento de texto manuscrito es la tarea de transcribir en un formato electrónico, datos escritos a mano (Vinciarelli, 2002). La importancia de esta tarea radica en recuperar información escrita a mano, de tal forma que se convierta en texto que pueda ser editado o procesado por una computadora.

Existen dos enfoques para llevar a cabo la tarea del reconocimiento de texto manuscrito (Plamodon & Srihari, 2000). El primero es el enfoque “en línea”, en el cual el reconocimiento se hace mientras el usuario escribe en algún dispositivo digital que permite a la computadora extraer características del texto escrito. Este tipo de enfoque tiene la ventaja de poder contar con información espacio-temporal que puede ser obtenida mientras se está escribiendo. Otro enfoque es el “fuera de línea”, en donde la entrada al reconocedor es un conjunto de imágenes que contienen texto. Esta tarea es más compleja que el reconocimiento en línea ya que no se tiene más información que la imagen, por lo que el preprocesamiento de la misma y la extracción de características adquieren un papel muy importante.

El reconocimiento de texto manuscrito normalmente se divide en diferentes etapas. En el sistema de reconocimiento de texto manuscrito antiguo llamado Priscus se identifican las etapas que se muestran en la Figura 2.1 (Gómez-Gil et al., 2007). De estas etapas destacan las siguientes: el preprocesamiento de la imagen, la segmentación, el reconocimiento de caracteres y el reconocimiento de palabras. Enseguida se describe brevemente cada una de éstas:

- Preprocesamiento. Su objetivo es preparar la imagen para que el proceso de segmentación y la extracción de características se lleve a cabo de forma más eficiente. En esta fase se busca eliminar el ruido de la imagen, es decir, los elementos que no son de utilidad para el reconocimiento del texto y aquellos que no forman parte del texto. Además, se busca normalizar las imágenes de tal forma que permitan extraer características similares para datos similares; esto se logra corrigiendo las variaciones en la inclinación de caracteres, altura de los trazos y ancho de las letras.
- Segmentación. Aquí se realizan las tareas necesarias para obtener de la imagen segmentos individuales que son de interés al reconocimiento. La segmentación puede ser a nivel de renglones, palabras o letras.
- Reconocimiento de caracteres. Consiste en clasificar el carácter contenido en una imagen como alguno de los posibles caracteres dentro de un alfabeto definido o dentro de algún grupo de segmentos generados de manera no supervisada.
- El reconocimiento de palabras consiste en encontrar dentro de las palabras de un léxico definido, la más compatible con los datos de la imagen (Vinciarelli, 2002). Existen dos principales categorías en las que se agrupan las técnicas de reconocimiento de palabras manuscritas (Namane et al., 2005):
  - Analíticas. En estas técnicas la palabra es segmentada en caracteres o pseudo-caracteres y entonces, cada uno de esos segmentos es identificado con modelos específicos. La ventaja de estas técnicas es que sólo se necesita una cantidad pequeña de modelos para identificar los pseudo-caracteres,

mientras que la principal desventaja es lidiar con el problema de la segmentación automática.

- Global. En estos métodos se busca identificar la palabra como si fuera una simple entidad, es decir, se extraen características de la imagen completa en sin considerar caracteres. Estas técnicas tienen como ventaja el no lidiar con el problema de segmentación, sin embargo, es necesario crear un modelo por cada palabra que se pretenda reconocer.

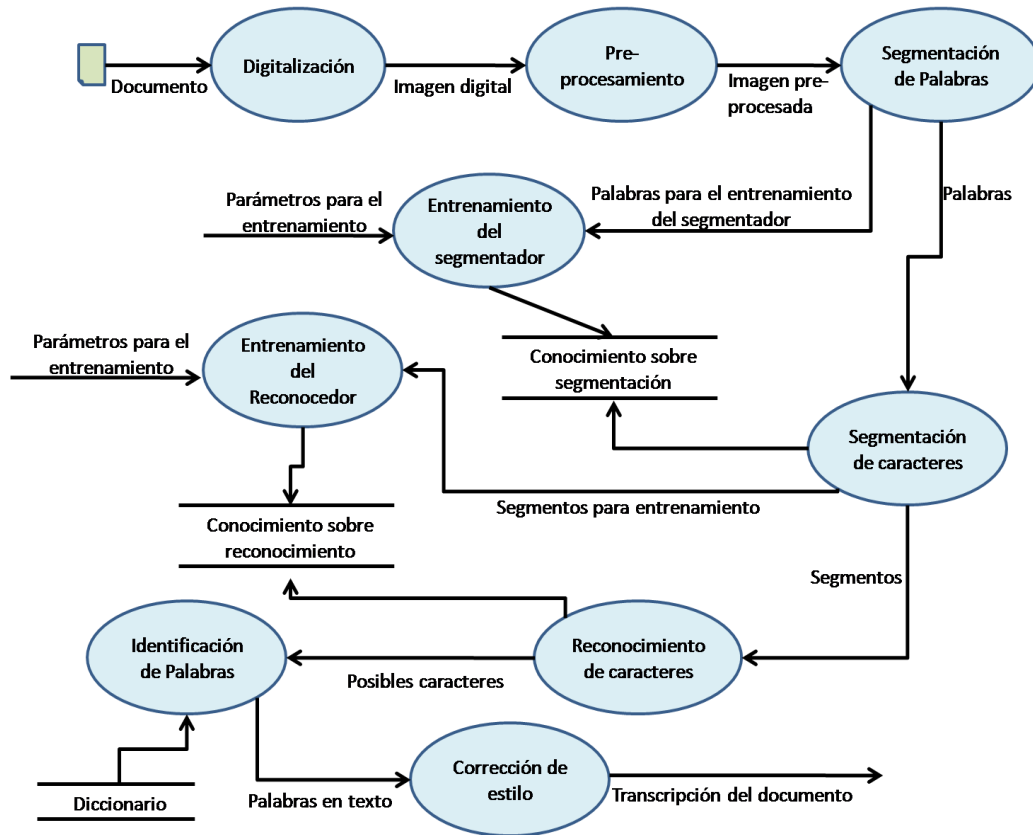


Figura 2.1 Etapas del proceso de reconocimiento de texto manuscrito propuesto por Gómez-Gil y colaboradores (2007).

## 2.2 Preprocesamiento de imágenes con texto manuscrito

El preprocesamiento de imágenes es una tarea importante y necesaria en el reconocimiento palabras. El preprocesamiento busca reducir las variaciones entre los diferentes estilos de escritura dependientes del escritor o de otros factores; además se

busca eliminar el ruido que pueda contener la imagen. Entre las variaciones que normalmente se toman en cuenta se encuentran la inclinación de los trazos (*slant*), el ancho de las letras y la inclinación de la línea base. Enseguida se explican detalladamente estos procesos.

### 2.2.1 Binarización de imágenes

La binarización de las imágenes consiste en convertir una imagen que puede contener píxeles con valores de intensidad o colores variados, en una imagen con sólo dos niveles de intensidad. Esta acción se logra al aplicar el operador umbral, el cual, para cada píxel de la imagen realiza una comparación de su valor de intensidad con respecto a un valor umbral; si lo excede el valor de intensidad del píxel es cambiado a 0 y si es igual o menor, entonces el nuevo valor de intensidad es 1. De esta forma resulta una imagen representada con una matriz binaria. La Figura 2.2 muestra una imagen en escala de grises y la imagen generada después de su binarización.



Figura 2.2 Ejemplo de binarización de una imagen. (a) imagen tomada de la base de datos IAM-DB (Martí & Bunke, 1999). (b) imagen binarizada.

### 2.2.2 Corrección de la inclinación

La inclinación de trazo o *slant* se define como el ángulo de un trazo con respecto a la vertical de la imagen. El texto manuscrito no siempre se presenta de forma vertical, sino que cada escritor acostumbra darle cierto grado de inclinación a éste (ver Figura 2.3). Asimismo, dentro de una palabra pueden presentarse varias inclinaciones. Existen varios métodos para tratar de eliminar esta inclinación, por ejemplo los descritos en (Toselli, 2004), (Ding et al., 2004), (Dong et al., 2005). En



general, la corrección de la inclinación se hace calculando el ángulo y aplicando una transformación de presión o empuje horizontal (*shear transformation*) (Shapiro & Stockman, 2001), la cual está definida por la siguiente operación:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -1 & sx \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.4)$$

donde  $(x', y')$  es la nueva coordenada al aplicar la transformación en el punto  $(x, y)$  y  $sx$  es la tangente del ángulo de inclinación de trazo.



**Figura 2.3** Ejemplo de una palabra con inclinación de trazo  $\alpha$  (imagen tomada de la base de datos IAM DB (Marti & Bunke, 1999)).

La diferencia entre los distintos métodos que buscan la corrección de la inclinación está en la manera de obtener el ángulo. El método propuesto por Toselli consiste en aplicar a la imagen original los operadores de Sobel horizontal y vertical (Shapiro & Stockman, 2001). Estos operadores calculan la diferencia en la intensidad de color en regiones locales de la imagen y están definidos de la siguiente forma:

$$G_{S_h} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.5)$$

$$G_{S_v} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Al aplicar los operadores de Sobel se obtiene una nueva imagen por cada operador:  $h(x, y)$  y  $v(x, y)$ ; con dichas imágenes se calcula el ángulo del gradiente de

la intensidad de color en cada punto de la imagen con respecto al eje vertical, con la siguiente expresión (Toselli, 2004):

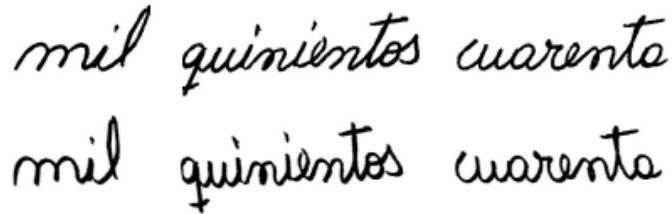
$$\text{angulo}(x, y) = 90 - \arctan\left(\frac{v(x, y)}{h(x, y)}\right) \quad (2.7)$$

Una vez calculados todos los ángulos, se toma la media de éstos como parámetro de inclinación dominante y es aplicada una transformación de empuje horizontal acorde a este parámetro. La Figura 2.4 muestra un ejemplo de corrección de inclinación sobre palabras manuscritas utilizando el método propuesto por Toselli.

Yimei Ding y colaboradores (2004) propusieron otro método de corrección de inclinación en el cual utilizan el código de cadena para representar la imagen de la palabra. El código de cadena es utilizado para representar los bordes de las formas contenidas en una imagen monocromática. A partir de un punto inicial se comienza a recorrer el borde guardando sólo la dirección hacia la cual se hizo el movimiento, generalmente se utilizan 8 direcciones (ver Figura 2.5) representadas por los números 0 al 7. Para estimar el ángulo promedio de inclinación de los trazos verticales con respecto a la coordenada horizontal  $x$  se utiliza la siguiente fórmula:

$$\theta(x) = \tan^{-1} \left[ \frac{n_1(x) - n_3(x)}{n_1(x) + n_3(x) + n_3(x)} \right] \quad (2.8)$$

donde  $n_i(x)$   $i = 1, 2, 3$  es la frecuencia de elementos en el código de cadena con dirección  $i \times 45^\circ$ .



mil quinientos cuarenta  
mil quinientos cuarenta

Figura 2.4 Ejemplo de corrección de inclinación (Toselli, 2004).

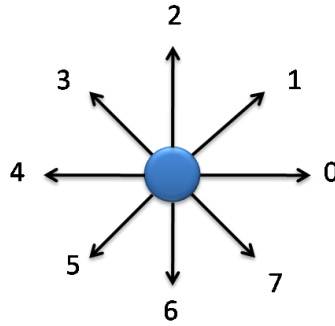


Figura 2.5 Direcciones utilizadas para la representación de imágenes con código de cadena (Ding et al., 2004).

Una vez obtenido el ángulo se aplica una operación de empuje horizontal para corregir la inclinación. La Figura 2.6 muestra el resultado de corregir la inclinación con este método.



Figura 2.6 Ejemplo de la corrección de inclinación (Ding et al., 2004).

La sección 4.1.1 describe el preprocesamiento utilizado para esta investigación el cual emplea el método de corrección de inclinación de trazo propuesto por (Ding et al., 2004).

### 2.3 Segmentación

La segmentación es la tarea de dividir la imagen que contiene una palabra en unidades básicas que son utilizadas en el proceso de reconocimiento (Vinciarelli, 2002). La segmentación puede ser explícita, si se requiere obtener caracteres, o

implícita, si no es necesario para el reconocedor utilizar caracteres. La segmentación explícita es una tarea muy complicada y puede implicar muchos errores debido a la paradoja de Sayre la cual afirma que un carácter no puede ser segmentado antes de ser reconocido y, a la vez, no puede ser reconocido sin antes haber sido segmentado. La segmentación implícita suele ser menos complicada ya que busca sobre segmentar las palabras sin buscar que los segmentos coincidan con caracteres, sin embargo, es necesario que los segmentos sean lo suficientemente consistentes para reconocedor.

## **2.4 Clasificación de secuencias temporales**

En el mundo real varios problemas de clasificación que se presentan no dependen solamente de las observaciones en un momento dado, sino que, para poder clasificar adecuadamente, se necesita utilizar información sobre el pasado. Por lo tanto, lo que se clasifica es una secuencia de eventos que han surgido uno tras otro, mas que eventos aislados. Una secuencia temporal consiste en una serie de componentes, que pueden ser escalares o vectores, y que presentan un orden inherente (Gupta et al., 2000). Ejemplos de este tipo de secuencias pueden ser los vectores de características obtenidos en las tareas reconocimiento de habla y reconocimiento de texto manuscrito, entre otros. La tarea de clasificar una secuencia temporal consiste en asignar la secuencia a una clase, teniendo en cuenta no sólo la clasificación que se le puede dar a cada uno de sus componentes, sino también el orden en el que aparecen éstos. Esta tarea resulta compleja debido a que varias secuencias pueden tener componentes similares y además al hecho de que en algunos casos una secuencia temporal puede ser parte de otra. El reconocimiento de palabras es considerado como un proceso de clasificación de secuencias temporales, en donde los componentes de la secuencia están formados por segmentos de la palabra (Graves et al., 2009). Actualmente existen varias metodologías para la clasificación de secuencias; en las siguientes secciones se describen las más utilizadas.

### 2.4.1 Técnicas para la clasificación de secuencias temporales

Para llevar a cabo la tarea de clasificación de secuencias temporales existen varias técnicas de las cuales destacan: los modelos ocultos de Markov (HMM por sus siglas en inglés) y las redes neuronales artificiales recurrentes.

#### Modelos Ocultos de Markov

Los Modelos Ocultos de Markov son modelos estocásticos de secuencias de observaciones; dichas observaciones son generadas por un sistema que puede cambiar de estado en pasos regulares (Vinciarelli, 2002). El conjunto de estados es finito y la transición de uno a otro depende de un proceso estocástico Markoviano (aquel en el que la probabilidad de transición al estado siguiente sólo depende del estado actual). Un modelo oculto de Markov consta de dos procesos estocásticos, uno que genera las observaciones en cada estado y otro que controla las transiciones entre estados, en el cual la secuencia de estados no es observable. Formalmente un HMM está definido como:

$$\lambda = \{Q, O, \pi, A, B\} \quad (2.9)$$

Donde  $Q$  es un conjunto finito de estados,  $O$  es el conjunto de observaciones que pueden ser generadas,  $\pi$  es el vector de probabilidades iniciales que contiene la probabilidad de cada estado de ser el primero en la secuencia,  $A$  es una matriz de probabilidades de transición entre estados y  $B$  es una matriz con la probabilidad para cada estado de generar una observación.

Existen tres problemas básicos que se pueden resolver con modelos ocultos de Markov (Rabiner & Juang, 1986):

- Evaluación. Dado un modelo, se calcula la probabilidad de que éste produzca una secuencia dada de observaciones.
- Secuencia óptima. Dado un modelo, se debe obtener la secuencia de estados más probable correspondiente a una secuencia de observaciones.

- Aprendizaje. Dada una secuencia de observaciones se ajustan los parámetros del modelo.

Los modelos ocultos de Markov han sido ampliamente utilizados en tareas de reconocimiento de habla (Chen & Chien, 2009) y reconocimiento de palabras (El-Yacoubi et al., 1999), entre otras tareas. En la sección 3.1 se describen brevemente los resultados obtenidos por reconocedores de palabras basados en HMM.

### **Redes Neuronales Recurrentes**

Una red neuronal artificial es un procesador distribuido, masivamente paralelo, compuesto de unidades simples de procesamiento, que puede almacenar conocimiento experimental y hacerlo disponible para su uso (Haykin, 1999). Las unidades simples de procesamiento pueden ser conectadas unas con otras; dichas conexiones tienen asociados pesos, los cuales son utilizados para almacenar el conocimiento adquirido. Las redes neuronales recurrentes son aquellas que poseen conexiones de retroalimentación entre unidades de procesamiento de manera que forman ciclos; dichas conexiones les proporcionan un comportamiento temporal dinámico. Las conexiones de retroalimentación también son consideradas memoria, la cual puede ser utilizada en la clasificación de secuencias temporales, debido a que pueden procesar secuencias arbitrarias de entrada. Una descripción general de las redes neuronales se da en la siguiente sección.

## **2.5 Redes Neuronales Artificiales**

Las redes neuronales artificiales son estructuras de procesamiento paralelo distribuido, algunas de estas estructuras fueron biológicamente inspiradas (Sierra Araujo, 2006), están formadas por unidades de cómputo simples no lineales a las que se da el nombre de neuronas, dichas neuronas se conectan entre sí y con el medio externo a través de enlaces conocidos como sinapsis que poseen un peso asociado. La Figura 2.7 muestra el modelo típico de una neurona artificial.

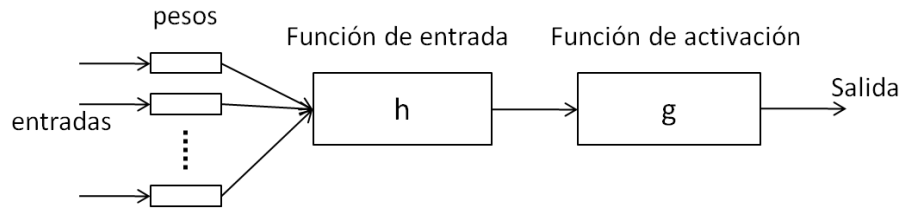


Figura 2.7 Modelo de una neurona artificial (Sierra Araujo, 2006).

Generalmente la operación de la neurona implica la evaluación de una función  $h$  a partir de las señales de entrada y el posterior cálculo de una función de activación  $g$ . De esta forma la salida de cada neurona depende de sus señales de entrada, los pesos asociados a sus sinapsis y las funciones de entrada y evaluación.

La Figura 2.8 muestra una red neuronal genérica, en la cual se puede observar que las neuronas pueden tener conexiones con cualquier otra, las señales de entrada pueden ser presentadas a cualquier neurona e incluso las salidas pueden tomarse de cualquier neurona. La forma en que se organizan e interconectan los nodos y las entradas se conoce como arquitectura de la red. La arquitectura de la red es muy importante, ésta define la forma en que las unidades comparten información, una red con conexiones que forman ciclos tienen un comportamiento dinámico, esto les permite utilizar información de estados anteriores de la red para procesar los nuevos datos de entrada, mientras que las redes que no tienen conexiones cíclicas tienen un comportamiento estático y funcionan como combinadores no lineales. A continuación se describen dos tipos de redes que fueron utilizadas en la tesis presentada en este documento.

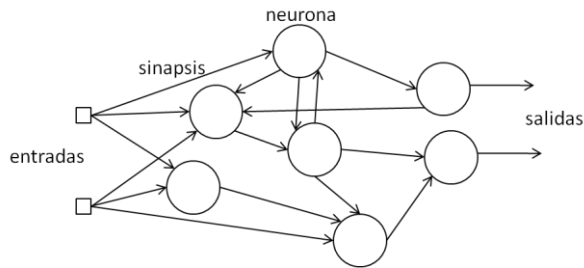


Figura 2.8 Red neuronal genérica (Sierra Araujo, 2006).

### **2.5.1 Red alimentada hacia adelante**

Una de las redes más utilizadas es la red alimentada hacia adelante (FF por sus siglas en inglés), en la arquitectura de esta red sólo se permiten conexiones hacia adelante, esto implica que no pueden existir conexiones que formen ciclos en la red; las neuronas se organizan por capas, por lo menos deben existir dos de ellas: la capa de entrada que corresponde a las entradas de la red y una capa de salida. Entre dichas capas pueden existir otras a las que se denominan capas ocultas. El entrenamiento de este tipo de redes consiste en ajustar los pesos de las sinapsis de tal forma que se minimice una función de error calculada a partir de la salida obtenida de la red y la salida esperada. El algoritmo básico de entrenamiento de las redes FF es conocido como algoritmo de retropropagación, el cual busca ajustar los pesos de la red teniendo como objetivo minimizar el error cuadrado obtenido al evaluar la red con todos los patrones del conjunto de entrenamiento. El algoritmo de retropropagación es descrito en el Apéndice A de este documento.

Estas redes neuronales artificiales son capaces de procesar patrones estáticos, es decir puede clasificar objetos que son descritos por un solo vector de características y no dependen del tiempo. Las redes FF son comúnmente utilizadas para la tarea de clasificación de patrones y aproximación de funciones.

### **2.5.2 Redes neuronales recurrentes**

Las redes neuronales recurrentes, a diferencia de las FF, permiten la existencia de ciclos formados por las conexiones, lo cual le brinda un comportamiento dinámico a la red. Dicho dinamismo permite modelar otros sistemas dinámicos dependientes del tiempo y son ampliamente utilizadas en la predicción de series de tiempo. Además, las conexiones recurrentes representan memoria en la red, guardando información de estados anteriores de la red, esto permite la clasificación de objetos que son representados por una secuencia de patrones que tienen relaciones



temporales. Existen varias arquitecturas de redes neuronales recurrentes que se han propuesto para diversas tareas. Algunas de las más utilizadas son:

- Red totalmente recurrente. Como el nombre lo indica en esta red todas las unidades de cálculo están interconectadas entre sí.
- Red de Jordan. Es una red con 3 capas que utiliza un conjunto de unidades de cálculo adicionales llamadas unidades de contexto, asociadas a la capa de salida. Estas unidades de contexto guardan una copia de las salidas en el tiempo  $t$  y estos datos son introducidos como entradas a las neuronas de salida en el tiempo  $t + 1$ . De esta forma se utiliza información de las salidas pasadas de la red.
- Red de Elman (Elman, 1990). También es conocida como red neuronal simple, al igual que la red de Jordan está compuesta por 3 capas y un conjunto de unidades de contexto, pero en esta arquitectura las unidades de contexto están asociadas a la capa escondida, guardando en cada paso información asociada a el estado interno de la red.

El algoritmo básico de entrenamiento de las redes recurrentes es una variación del algoritmo de retropropagación y es conocido como retropropagación a través del tiempo. Dicho algoritmo busca desdoblarse la red de tal forma que se obtenga una estructura similar a la de una red FF y así poder aplicar el algoritmo básico de retropropagación. El algoritmo se describe detalladamente en el Apéndice B.

## 2.6 Mapas Auto-Organizados

Los mapas auto-organizados (SOM por sus siglas en inglés) son redes neuronales artificiales que fueron introducidas por Teuvo Kohonen (Kohonen, 1981), su arquitectura consta de 2 capas, una de entrada y una de salida, en donde la capa de salida normalmente es un arreglo bidimensional de neuronas, al cual se le conoce como mapa topográfico (ver Figura 2.9). La red SOM está inspirada en teorías biológicas que indican que distintas áreas del cerebro se especializan en diferentes tareas como: el habla, el movimiento, la visión, planeación, etcétera. Por lo tanto,

para actividades diferentes se activan zonas de la corteza cerebral distintas. Las redes SOM imitan dicha característica del cerebro, buscando que datos diferentes introducidos en la red activen zonas distintas del mapa que forman las neuronas de salida.

La red SOM cuentan con 2 características muy importantes, una de ellas es la auto-organización, la cual se logra durante la fase de entrenamiento al adaptarse los pesos de las neuronas de salida de tal forma que para dos patrones de entrada similares que se encuentran cercanos en el espacio de entrada, la activación que producen en el mapa de salida SOM es también cercana (Kangas, 1994). Otra característica importante es que preservan, en el mapa bidimensional, las relaciones métricas y/o topológicas que existen entre los datos de entrenamiento. Los mapas auto-organizados son formalmente descritos por Kohonen (Kohonen, 2001) como un mapeo no lineal y ordenado de datos de entrada de altas dimensiones a elementos de un arreglo de baja dimensión.

Los mapas auto-organizados se entrenan utilizando un algoritmo de aprendizaje no supervisado y competitivo. El aprendizaje no supervisado es un método en el cual no se tiene conocimiento *a priori* de las clases a las que pertenecen los objetos que son utilizados para el entrenamiento del sistema. Este tipo de métodos generalmente construyen grupos de objetos parecidos (*clusters*), en donde el parecido está asociado a una medida de similitud entre objetos. En redes neuronales, el aprendizaje competitivo se da cuando las neuronas de la capa de salida compiten entre ellas por ser activadas (Haykin, 1999); la neurona ganadora, o con mayor activación, es aquella que se asemeja más al patrón de entrada. El aprendizaje consiste en reforzar las conexiones de la neurona ganadora y debilitar las otras, de tal forma que los pesos de la unidad ganadora se asemejen cada vez más a los datos de entrada.

El algoritmo propuesto por Kohonen es un proceso que obtiene como resultado un mapa ordenado, en el cual para ciertos patrones de entrada las neuronas con mayor activación corresponden a cierta área del mapa.

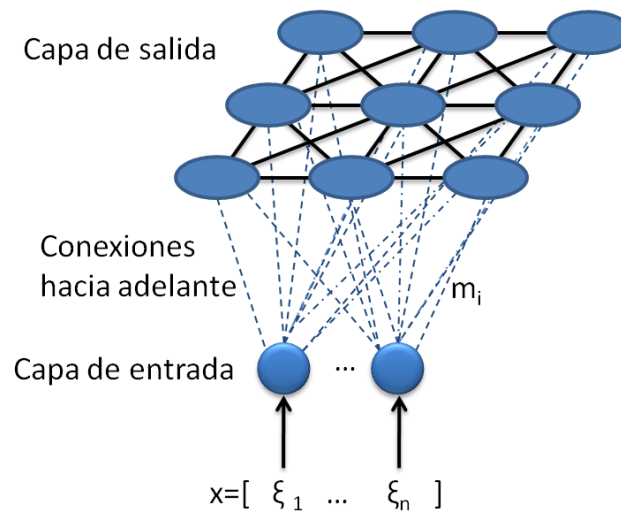


Figura 2.9 Arquitectura de la Red SOM (Kohonen, 2001).

El algoritmo básico de entrenamiento de la red SOM (Kohonen, 1981) se describe a continuación:

- 1) Cada neurona  $i$  de la capa de salida tiene asociado un vector de pesos  $m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathfrak{R}^n$ , también conocido como vector de referencia,  $n$  es la dimensión de los datos de entrada. Cada elemento  $m_i$  es inicializado con un valor aleatorio.
- 2) Cada patrón de entrada  $x = [\xi_1, \xi_2, \dots, \xi_n] \in \mathfrak{R}^n$  del conjunto de entrenamiento, es comparado con cada neurona  $m_i$  del mapa. Esta comparación generalmente se hace utilizando la distancia Euclidiana  $\|x - m_i\|$ . La neurona ganadora  $c$  es aquella que tiene la menor distancia con respecto al patrón de entrada, esto es:

$$c = \operatorname{argmin}_{m_i} \{\|x - m_i\|\} \quad (2.2)$$

- 3) Una vez seleccionada la neurona ganadora, la actualización de los vectores de peso se hace de acuerdo a la siguiente expresión:

$$m_i(t + 1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \quad (2.3)$$

Donde  $h_{ci}$  es una función, conocida como función de vecindad, que controla el refuerzo que se da a cada uno de los vectores de pesos durante la actualización, de tal forma que una neurona cercana a la ganadora tenga mayor refuerzo que una lejana. Para lograr que el algoritmo converja es necesario que  $h_{ci} \rightarrow 0$  cuando  $t \rightarrow \infty$ . Una función de vecindad comúnmente utilizada es:

$$h_{ci} = \alpha \cdot \exp\left(\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right) \quad (2.4)$$

Donde  $\alpha$  es un valor escalar que decrece con el tiempo (conforme  $t$  aumenta) y  $\sigma(t)$  define el ancho de la vecindad en la iteración  $t$ ;  $r_c$  son las coordenadas de la neurona ganadora y  $r_i$  son las coordenadas de la  $i$ -ésima neurona del mapa de salida de la red SOM.

# Capítulo 3

## Trabajo relacionado

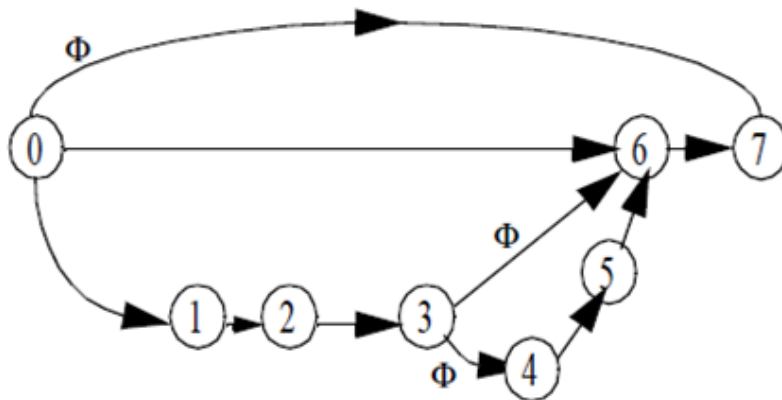
Los métodos para el reconocimiento de palabras manuscritas fuera de línea suelen dividirse en dos categorías: analíticos y globales (ver sección 2.1). Las herramientas más utilizadas en los métodos de reconocimiento de texto manuscrito han sido los HMM y las redes neuronales artificiales. En este capítulo se describen algunos de los trabajos más relevantes relacionados con el reconocimiento de palabras utilizando las que se han realizado en los últimos años.

### 3.1 Trabajos que utilizan modelos ocultos de Markov

Los HMM son utilizados en el reconocimiento de palabras bajo la suposición de que una imagen de palabra puede representarse como una secuencia de observaciones. Por lo tanto, se utilizan los HMM para generar el modelo de cada palabra de un léxico definido buscando que las observaciones que pueden ser producidas en cada estado del modelo correspondan a segmentos de la imagen que se está analizando. En general los trabajos reportados construyen los modelos de cada palabra a partir de la concatenación de modelos correspondientes a los caracteres que la forman. Por lo tanto, para modelar cada palabra se debe tener el número de estados necesarios para modelar cada uno de los caracteres.

La diferencia entre los distintos trabajos basados en HMM y reportados recientemente radica en el número de estados del modelo utilizado y las transiciones entre ellos, además de las características que son extraídas de las imágenes y dadas

como secuencia de entrada. Cabe recordar que para proponer un modelo generalmente se hacen suposiciones sobre la estructura que éste debería tener. El-Yacoubi y colaboradores (1999) basan las suposiciones para proponer un modelo en los resultados de su segmentador. Éste puede generar caracteres correctamente segmentados, omitir un carácter o sobre-segmentar en dos o tres partes. Para modelar este tipo de caracteres o pseudo-caracteres, los autores proponen un HMM de 7 estados, mostrado en la Figura 3.1. La transición del estado 0 al estado 7 (representado como  $t_{07}$ ) emite el símbolo nulo  $\phi$  modelando el caso de omitir un carácter; la transición  $t_{06}$  emite un símbolo que codifica a un carácter correctamente segmentado; la transición nula  $\phi$   $t_{36}$  modela el caso de sobre-segmentación en solo 2 segmentos. Las transiciones  $t_{01}$ ,  $t_{23}$  y  $t_{56}$  están asociadas a la primera, segunda y tercera parte de la sobre-segmentación, mientras que las transiciones  $t_{12}$  y  $t_{45}$  modelan la naturaleza de los puntos de segmentación que dan lugar a la sobre-segmentación. Una vez entrenados los modelos para cada carácter, el modelo de la palabra se obtiene al concatenar los HMM's correspondientes a los caracteres que componen la palabra. El método propuesto por El-Yacoubi fue probado por una base de datos compuesta por nombres de ciudades francesas; la Tabla 3.1 muestra los resultados obtenidos. Las características extraídas de las imágenes los trazos ascendentes y descendentes, los trazos con huecos, y los histogramas horizontal y vertical de cada carácter segmentado.

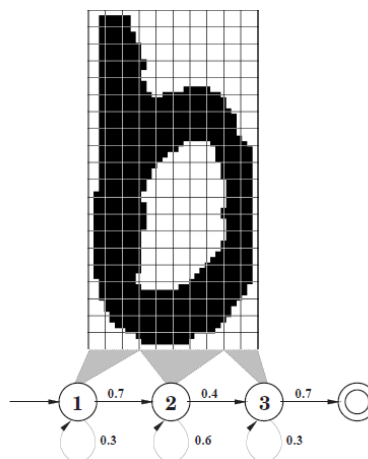


**Figura 3.1** Modelo oculto de Markov de 7 estados propuesto por El-Yacoubi y colaboradores (1999), para el modelado de un carácter.

Alejandro Toselli (2004) utiliza 3 características obtenidas de imágenes que contienen texto: la derivada horizontal, la derivada vertical y el nivel de gris de cada una de las columnas de la imagen. En este trabajo se utilizaron diferentes modelos de Markov para cada uno de los símbolos del alfabeto, tratando de que cada estado modele columnas de la imagen de un carácter con características similares. La Figura 3.2 muestra un ejemplo del modelo para la letra *b* en el cual un estado modela secciones similares de la imagen; el estado 1 modela las columnas con todos o casi todos sus píxeles negros, el estado 2 modela columnas con píxeles negros en la zona media y baja de la imagen, separados por píxeles blancos; y por último el estado 3 modela las columnas con píxeles negros de la parte media de la imagen hacia abajo. Las áreas grises que unen los estados con la imagen del carácter indican las columnas de la imagen que son representadas por cada estado.

**Tabla 3.1. Porcentaje de reconocimiento obtenido en el trabajo de El-Yacoubi y colaboradores (1999).**

Tamaño del léxico	Porcentaje de Reconocimiento
10	99.0%
100	96.1%
1000	87.7%



**Figura 3.2 Modelo de Markov de 3 estados que modela la letra manuscrita "b". Los números sobre los vértices indican la probabilidad de transición de un estado a otro (Toselli, 2004).**

A diferencia del trabajo de El-Yacoubi y colaboradores (1999), los cuales modelaron cada una de las palabras como la concatenación de los modelos de cada uno de los caracteres que la componían, Toselli hace el modelo del léxico utilizando autómatas estocásticos. La Figura 3.3 muestra un autómata estocástico para modelar las palabras “Mil” y “mil”. En estos autómatas cada uno de sus vértices es reemplazado por el HMM que modela al símbolo requerido para realizar la transición de un estado a otro (Figura 3.4). La ventaja de utilizar autómatas en combinación con los HMM es que se aprovecha el modelo de algún carácter para las diferentes palabras que lo contengan, siempre y cuando el autómata lo permita. Además, se pueden modelar frases completas con los autómatas, lo cual es muy útil en tareas que no tienen un léxico muy extenso.

Los experimentos realizados por Toselli se apoyan en una base de datos compuesta por nombres de cantidades numéricas, obtenidos de cheques bancarios. Se evaluó el método utilizando distintos números de estados en los HMM que modelan los caracteres. El tamaño del léxico utilizado fue de 52 palabras, el mejor resultado reportado fue 91.5% de precisión en el reconocimiento de palabras.

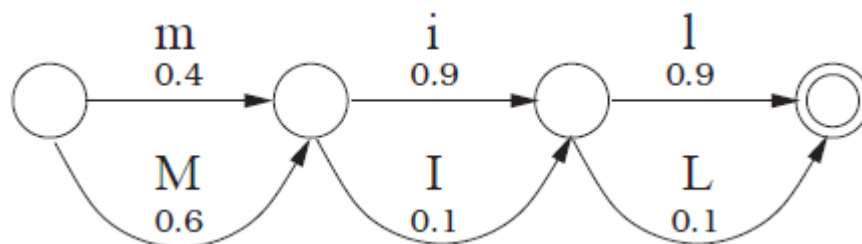


Figura 3.3 Autómata estocástico de estados finitos que modela la palabra "mil" y "Mil" (Toselli, 2004).

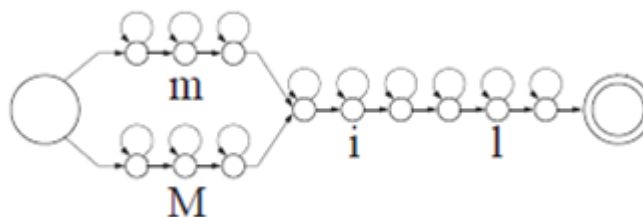


Figura 3.4 Autómata estocástico de estados finitos con transiciones reemplazadas por los modelos de cada carácter (Toselli, 2004).



### 3.2 Trabajos relacionados basados en redes neuronales artificiales

Graves y colaboradores (2009) proponen un método basado en un tipo novedoso de redes neuronales recurrentes diseñadas específicamente para el etiquetado de secuencias, donde los datos son difíciles de segmentar y tienen dependencias bidireccionales, es decir, la clasificación de la secuencia necesita de información contextual pasada y futura para su correcta clasificación. En este método, al igual que en los basados en HMM descritos anteriormente, los vectores que describen a la imagen están formados por características extraídas de cada columna de la imagen. La arquitectura de la red neuronal consta de una capa de entrada y una capa oculta denominada *Bidirectional Long Short-Term Memory* (BLSTM). Dicha capa está construida a partir de subredes conocidas como *Long Short-Term Memory* (LSTM), que, a diferencia de las redes neuronales recurrentes convencionales, no tiene el problema de conocido como “fuga del gradiente”, el cual hace difícil para una red recurrente recordar información relevante por más de 10 pasos en el tiempo; Hochreiter y colaboradores (2001) hacen un análisis profundo sobre este problema. Además, la capa BLSTM se divide en dos grupos de subredes para poder acceder a la información contextual en ambas direcciones a lo largo de la secuencia de entrada. El modelo utiliza como capa de salida la propuesta por Graves y colaboradores (2006) denominada *Connectionist Temporal Classification* (CTC), que está específicamente diseñada para la tarea de etiquetado de secuencias. El número de neuronas que conforman la capa CTC es igual al número de etiquetas de clase, más una adicional para representar blancos; las salidas de la capa son normalizadas con la función *softmax*, lo que implica que la suma de todas ellas es igual a 1 y cada una está en el rango (0,1]. La Figura 3.5 muestra el sistema de reconocimiento propuesto; la entrada al sistema es la imagen de texto manuscrito ilustrada en la parte inferior de la figura, dicha imagen es procesada y se le extraen características que posteriormente son evaluadas en la capa BLSTM para finalmente ser etiquetadas por la capa de salida CTC.

Murillo Gil (2007) propone un reconocedor de palabras basado en una red SOM que reconoce caracteres y un conjunto de autómatas finitos deterministas que representan cada una de las palabras que el sistema puede reconocer; este reconocedor trabaja con caracteres segmentados manualmente. Al igual que los otros trabajos expuestos, la primer etapa del reconocedor es el pre-procesamiento, el cual está basado en el propuesto por De los Santos (2003) que consiste en la eliminación de ruido, la normalización del tamaño de los caracteres y la corrección de la inclinación; ésta última se logró utilizando un algoritmo de rotación de la imagen después de calcular el ángulo de inclinación de trazo. La siguiente etapa en el reconocedor propuesto por Murillo Gil, es el reconocimiento de caracteres, el cual se lleva a cabo utilizando una red SOM, previamente entrenada para una muestra de caracteres. El mapa generado con la red SOM fue etiquetado automáticamente haciendo el recuento de los caracteres del conjunto de entrenamiento que activaron algún nodo después de haber entrenado la red. Para cada nodo el carácter que en más ocasiones lo activó se toma como la etiqueta más probable; sin embargo, se guarda un registro de la frecuencia con de la activación que tuvo para otros caracteres.

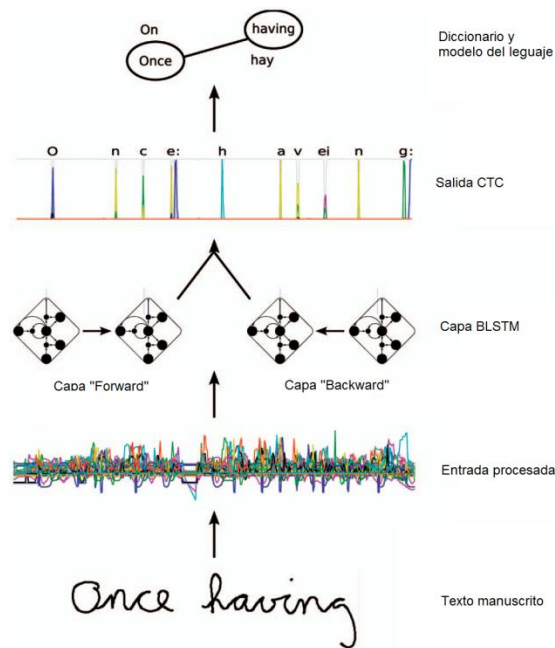


Figura 3.5 Modelo propuesto por Graves y colaboradores (2009)

Finalmente, la etapa de reconocimiento de palabras utiliza un autómata finito determinista para modelar cada palabra. La entrada a estos autómatas es el conjunto de etiquetas asignadas a una secuencia de caracteres que fueron procesados por la red SOM. Para dar soporte a los errores en el reconocimiento de caracteres, si el carácter reconocido no es alguno de los necesarios para la transición entre los estados del autómata, se toma la siguiente etiqueta que activó con más frecuencia al nodo. El proceso anterior se realiza para todas las opciones posibles, antes de ser rechazada una palabra. La palabra reconocida es aquella en la cual un autómata llegó a su estado final. La Figura 3.6 muestra el proceso para el reconocimiento de la palabra “casa”. Para probar el método Murillo Gil realizó pruebas para 40 palabras diferentes, obteniendo un porcentaje de reconocimiento de 65.85 %.

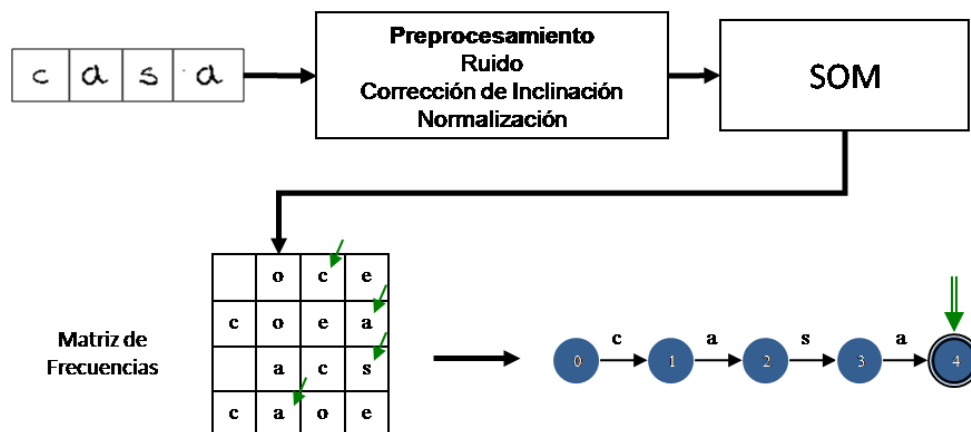
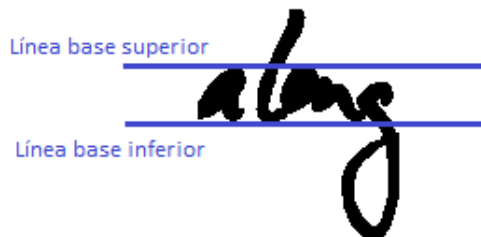


Figura 3.6 Proceso completo para el reconocimiento de una palabra (Murillo Gil, 2007).

Côté y colaboradores (1998) presentan un modelo de reconocimiento de palabras proponiendo una arquitectura inspirada en el comportamiento humano y en conceptos perceptuales. El método que propusieron se compone de 4 módulos: preprocesamiento, extracción de las líneas base, extracción de características y reconocimiento. El preprocesamiento consiste en la extracción de contornos, detección de ciclos e identificación de mínimos locales. La extracción de las líneas base consiste en dividir la imagen en tres regiones: una en donde se localizan los trazos ascendentes, otra para el cuerpo de la palabra y por último la región en donde

se localizan los trazos descendentes. Las líneas base se ubican en la parte superior e inferior del cuerpo de la palabra (Figura 3.7).



**Figura 3.7 Líneas base en una palabra.**

En el método propuesto por Côté y colaboradores se extraen tres tipos de características: primarias, secundarias y valles. Las características primarias son las ubicaciones de los caracteres clave dentro de la imagen, los caracteres clave son aquellos que contienen trazos ascendentes, descendentes o ciclos. La justificación de localizar estos caracteres clave es que los humanos los utiliza para reconocer una palabra, sin necesidad de reconocer todas las letras. Las características secundarias son condicionales, sólo son extraídas en caso de encontrar un trazo ascendente o descendente. Estas características consisten en indicar si a la derecha del trazo ascendente existe un ciclo, si el ciclo está a la izquierda del trazo ascendente debe corresponder a una “d” y si es a la derecha corresponde a una “b”. De igual manera cuando se detecta un trazo descendente el encontrar ciclos a la izquierda o derecha del mismo puede indicar que se encontró una letra “p” o “q”. Las últimas características que utilizan son los valles, primero se extrae el fondo de la imagen, entonces, los valles son las componentes conexas sobre el contorno superior de la palabra y debajo del contorno inferior de la misma.

El reconocimiento se lleva a cabo con una red neuronal que consta de tres niveles de neuronas ordenados de manera jerárquica: nivel de características, nivel de letras y nivel de palabras. Se supone que debe haber una neurona por cada palabra, letra y característica asociada a una posición en la imagen. Existen conexiones bidireccionales entre los niveles adyacentes, además, las celdas están conectadas de

acuerdo a conocimiento *a priori*. Por ejemplo la palabra *two* debe tener una conexión con la letra *t*, la letra *w* y la letra *o* los cuales son nombrados como vecinos. Las neuronas pueden estar activas o pasivas y su activación tiene un valor entre 0 y 1. Cuando una neurona es estimulada aumenta su activación y puede influenciar a sus vecinos.

Existen dos procesos que permiten la transmisión de información entre capas: el proceso *bottom-up* y el proceso *top-down*. El proceso *bottom-up* comienza en la extracción de características de una imagen que contiene una palabra, las características son introducidas a la capa de características, la cual le da prioridad a los ascendentes y descendentes, debido a que son las características más estables. Además, se localiza la posición relativa de estos trazos en la palabra. De esta forma las neuronas correspondientes a trazos ascendentes y descendentes en determinadas regiones de una palabra son estimuladas y propagan este estímulo a la capa de letras. De la misma manera, las letras estimuladas activan las neuronas a nivel de palabras con las que están vinculadas. Por lo tanto, al final de esta etapa solo algunas neuronas, en la capa de palabras, están activadas y otras no. Durante el proceso *top-down* se emplea la información contextual. Primero se alimenta hacia atrás la activación de las neuronas de la capa de palabras, estimulando a las neuronas en la capa de caracteres activadas anteriormente. Además, a partir de las neuronas activadas en la capa de palabras, se crea una hipótesis sobre que letras pueden estar junto a los trazos ascendentes o descendentes. Si esta hipótesis coincide con las características obtenidas en la imagen, entonces las neuronas correspondientes son estimuladas, de lo contrario la hipótesis es rechazada. A la tarea de llevar a cabo los dos procesos antes descritos se le denomina ciclo perceptual, el cual es ejecutado varias veces hasta que el sistema converja a una solución. Las palabras que resulten con mayor activación son seleccionadas como el resultado del reconocimiento. La Figura 3.8 muestra un esquema general del proceso de reconocimiento propuesto por Côté y colaboradores.

Para realizar pruebas se utilizaron 32 palabras distintas compuestas de 3 a 9 letras obtenidas de la base de datos CENPARMI. En la Tabla 3.2 se muestra el porcentaje de palabras correctamente clasificadas; N es el número de palabras candidatas resultantes del reconocimiento.

Tabla 3.2 Porcentaje de reconocimiento tomando N palabras candidatas (Côté et al., 1998).

N	1	2	5	10
Conjunto de Entrenamiento	76.1	88.0	91.8	94.0
Conjunto de Prueba	73.6	81.0	89.4	92.7

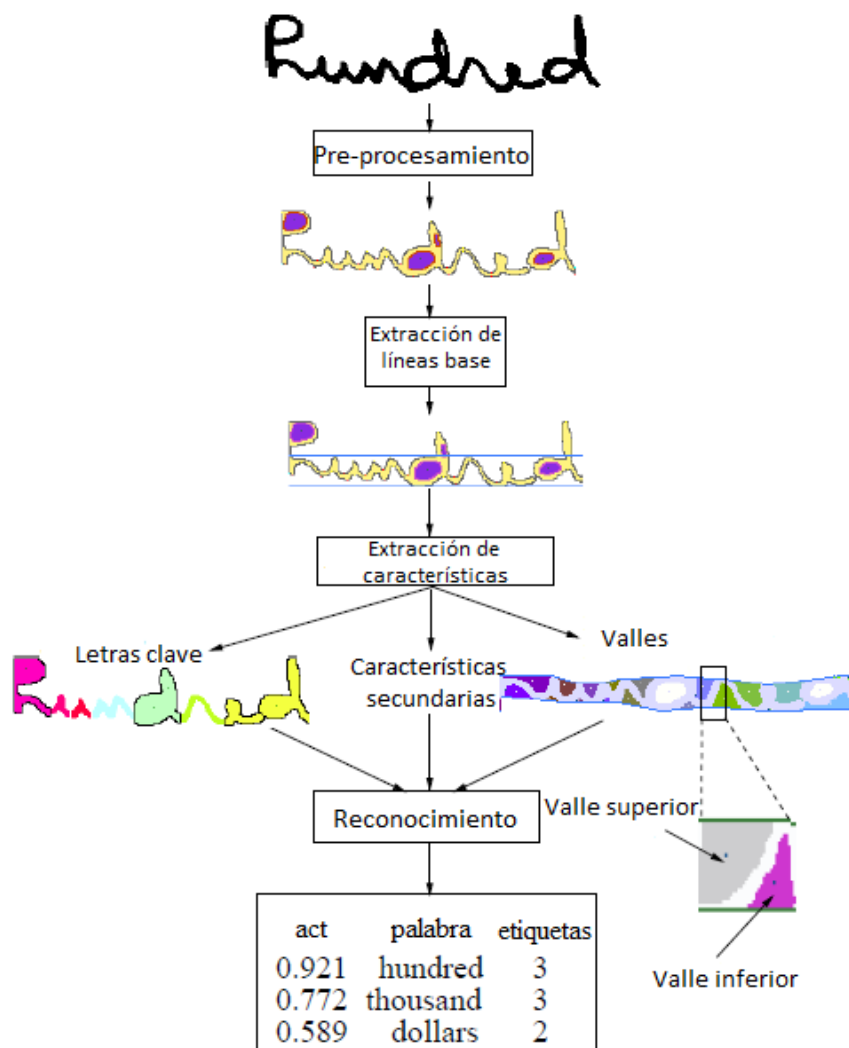


Figura 3.8 Esquema general del método propuesto por Cote y colaboradores (1998).

Namane y colaboradores (2005) proponen un método global de reconocimiento de palabras, usando cantidades escritas en cheques, que analiza la imagen de forma global, es decir, la imagen que contiene una palabra aislada es procesada sin ser segmentada. El sistema que proponen consta de dos fases: pre-procesamiento y reconocimiento.

La fase de pre-procesamiento consiste de suavizado, binarización, corrección de inclinación de trazos y de línea base, además de la normalización del tamaño de la imagen y la localización de tres zonas que para este trabajo son de interés: central, alta y baja. Las palabras manuscritas representan 27 cantidades distintas agrupadas en 16 prototipos creados al superponer 4 palabras de cada clase, el prototipo creado es almacenado en un tipo de red neuronal conocida como red de Hopfield. Utilizando de estos prototipos se entrena una red *feed-forward* para decidir a qué palabra corresponde. La Figura 3.9 muestra el sistema propuesto por Namane y colaboradores. El sistema logró un porcentaje de reconocimiento del 80.75 % para 27 palabras distintas.

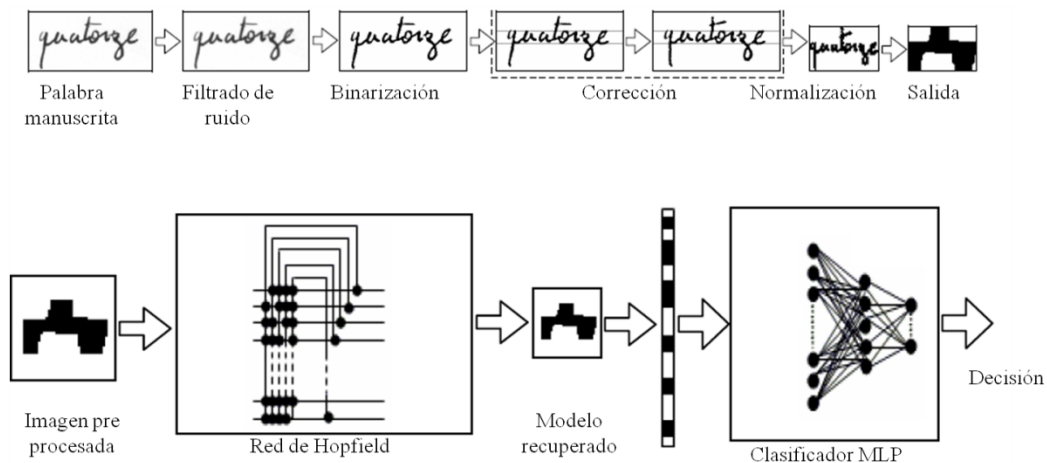


Figura 3.9 Sistema para el reconocimiento de palabras propuesto por Namane y colaboradores (2005).





# Capítulo 4

## Método Propuesto

En este capítulo se presenta una descripción del método propuesto para el reconocimiento de palabras manuscritas y las etapas que lo componen. También, se describe el proceso de entrenamiento de las redes neuronales que están involucradas en el método de reconocimiento.

### 4.1 Descripción General

El método propuesto se basa en el reconocimiento de secuencias temporales, en donde las secuencias corresponden a segmentos de imágenes que forman palabras y que han sido etiquetados con respecto a su probabilidad de pertenencia a un grupo específico, previamente generado. Para llevar a cabo el reconocimiento de palabras manuscritas se realizan algunas de las tareas comunes mencionadas en la sección 2.1 como el preprocesamiento de las imágenes, segmentación y reconocimiento de palabras.

El método propuesto involucra una tarea intermedia entre la obtención de los segmentos y el reconocimiento de palabras, la cual consiste en generar prototipos de los segmentos utilizando una red SOM. Las característica de esta red permite aportarle al clasificador información no sólo del prototipo que represente al segmento de entrada sino información del prototipo más parecido y los más similares a éste. De esta forma se pretende dar más información al clasificador para poder lidiar con el

problema del parecido entre segmentos y así mejorar el reconocimiento a nivel de palabras.

Las diferentes etapas del método propuesto se muestran en la Figura 4.1 y son descritas en las siguientes secciones del documento. Asimismo, se describe detalladamente el proceso de entrenamiento de la red neuronal utilizada para clasificar.

#### **4.1.1 Preprocesamiento**

El preprocesamiento empleado en el método propuesto es muy sencillo debido a que se busca simpleza en el método y además, dado que el método busca reconocer palabras sin ninguna restricción de estilo (cursiva o script), cualquier tipo de preprocesamiento tiende a fallar con algún o algunos estilos de escritura sin importar que tantas operaciones en su preprocesamiento se realicen. El preprocesamiento utilizado para el método que se propone consta de los siguientes pasos:

- a) Binarización de imágenes. Se obtienen imágenes binarias utilizando el método descrito en la sección 2.2.1. el cual está basado en el operador umbral. El operador umbral consiste en cambiar el valor de los valores de intensidad de los píxeles de la imagen, si el valor de intensidad es igual o mayor a un valor umbral, el nuevo valor de intensidad para ese píxel es el máximo, de lo contrario es el mínimo, de esta forma se obtiene una imagen en blanco y negro.
- b) Corrección de inclinación. La corrección de la inclinación de los trazos con respecto al eje vertical de la imagen se consigue aplicando el método propuesto por Ding y colaboradores (2004). Este método se describe en la sección 2.2.2.
- c) Después de la binarización y la corrección de la inclinación se eliminan los píxeles blancos que rodean a la palabra. Esto se logra eliminando las columnas y filas que no contienen ningún píxel negro, por lo tanto no tiene información útil para el reconocimiento.

En el apéndice C se muestran algunos de los algoritmos involucrados en el preprocesamiento.

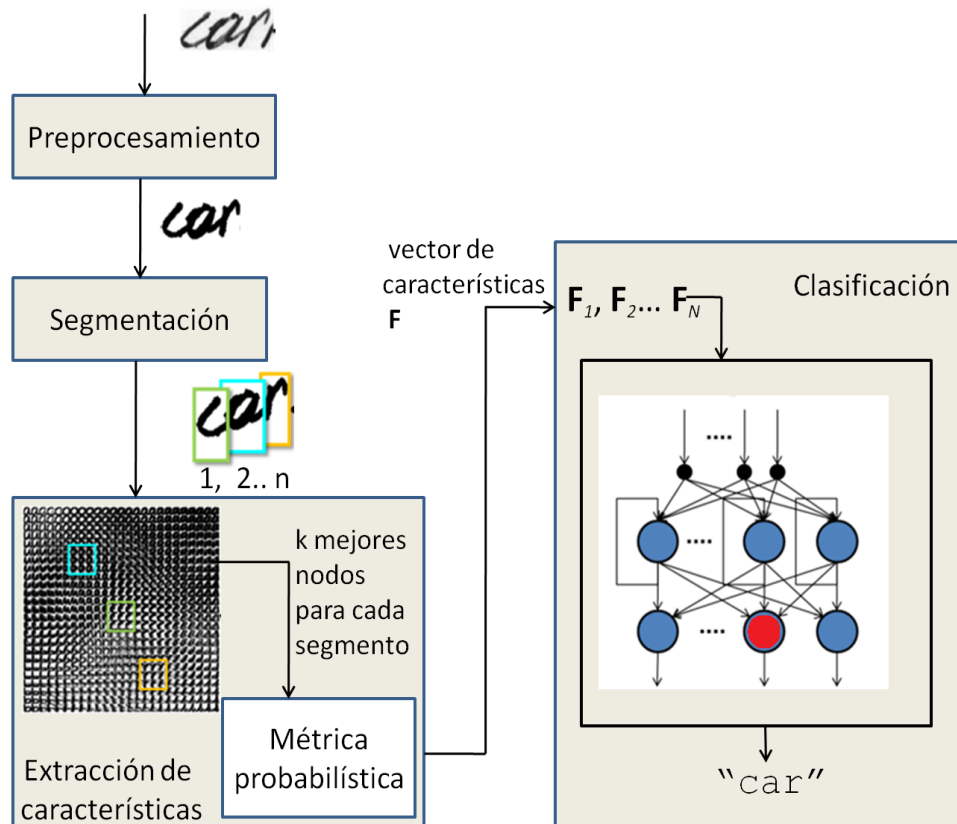
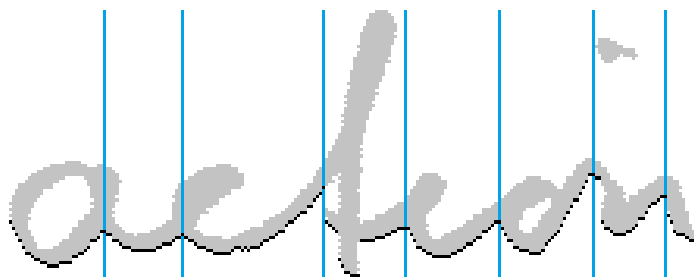


Figura 4.1 Método propuesto para la clasificación de una palabra.

#### 4.1.2 Segmentación de las imágenes

La segmentación de las imágenes consiste en dividir cada imagen que contiene una palabra en sub-imágenes que contengan los caracteres o pseudo-caracteres que forman a la misma. Cabe hacer notar que la segmentación de caracteres es a la fecha un problema abierto en el cual se ha trabajado ampliamente (Casey & Lecolinet, 1996).

Para el método propuesto en este trabajo de investigación no es de interés segmentar una palabra en caracteres, sino que para diferentes imágenes que contengan la misma palabra se generen segmentos similares. Para lograr esto se propuso un método de segmentación en el cual se buscan en cada una de las columnas de la imagen el pixel negro inferior. Una vez localizados estos pixeles se selecciona como punto de corte la ubicación de aquellos que se encuentran en una posición más alta que sus vecinos. La Figura 4.2 muestra un ejemplo de la segmentación generada con la heurística descrita anteriormente, los pixeles pintados de negros corresponden a los pixeles inferiores de cada columna y las líneas verticales marcan los puntos de segmentación. Finalmente, se eliminan las áreas blancas que rodean al segmento y son normalizados en tamaño escalándolos a 20x20 pixeles. El Apéndice D muestra el algoritmo completo de segmentación.



**Figura 4.2 Segmentación automática propuesta. La palabra se muestra en gris para resaltar los pixeles negros inferiores de cada columna. Las líneas verticales azules marcan los puntos de segmentación.**

### **4.1.3 Agrupamiento y extracción de características**

Al finalizar el preprocesamiento se tienen segmentos de imágenes de palabras, dichos segmentos tienen formas variables incluso para segmentos generados para una misma palabra. Además, utilizar la imagen como dato de entrada para un clasificador no es lo más conveniente debido a la dimensión de los datos. Para soportar la variabilidad en la forma de los segmentos una red SOM es entrenada creando prototipos de los segmentos, de esta forma los datos de entrada del el reconocedor de palabras estarán limitados a los prototipos generados en el mapa auto-organizado. Por

otra parte, una de las características principales de las redes SOM es que los prototipos generados mantienen la misma distribución dentro del mapa que la que tiene el conjunto de entrenamiento, debido a esto y a la organización que tienen los prototipos, es posible utilizar las coordenadas dentro del mapa para representar al segmento en lugar de utilizar la imagen binaria.

El uso de la red SOM tiene otra función importante en el método que se propone. Al generar un mapa que tiene una relación de semejanza entre los prototipos generados, permite suponer que si un segmento al ser clasificado utilizando el mapa no cae dentro del grupo al que realmente corresponde, entonces, alguno de los vecinos debiera de ser el grupo correcto. Bajo esta suposición podemos dar al reconocedor de palabras más información, tomando en cuenta a los vecinos del grupo al cual la red SOM asignó a el segmento.

En la primera etapa de entrenamiento del sistema, los segmentos de palabras obtenidos de las imágenes del conjunto de entrenamiento son utilizados para entrenar una red SOM con un nivel de salida de  $n \times m$  neuronas;  $n$  y  $m$  seleccionados experimentalmente, a fin de que se generen grupos de segmentos de palabra parecidos (la sección 2.6 describe el algoritmo de entrenamiento de la red SOM). Dado que este agrupamiento es no supervisado y no se tienen etiquetas para los segmentos generados, se mantiene la incertidumbre de los datos pero se establece una relación de similitud entre ellos representada por las relaciones topográficas de las neuronas del mapa SOM. La Figura 4.3 muestra un ejemplo de un mapa de salida de una red SOM, generado al ser entrenado con 12 clases de caracteres distintos (a, b, c, d, e, f, n, o, r, s, t, u) segmentados manualmente. Este ejemplo se muestra solamente para clarificar el tipo de agrupamiento que SOM genera. El mapa utilizado en el método propuesto se entrena a partir de segmentos de palabra obtenidos automáticamente, por lo que pueden ser o no caracteres.

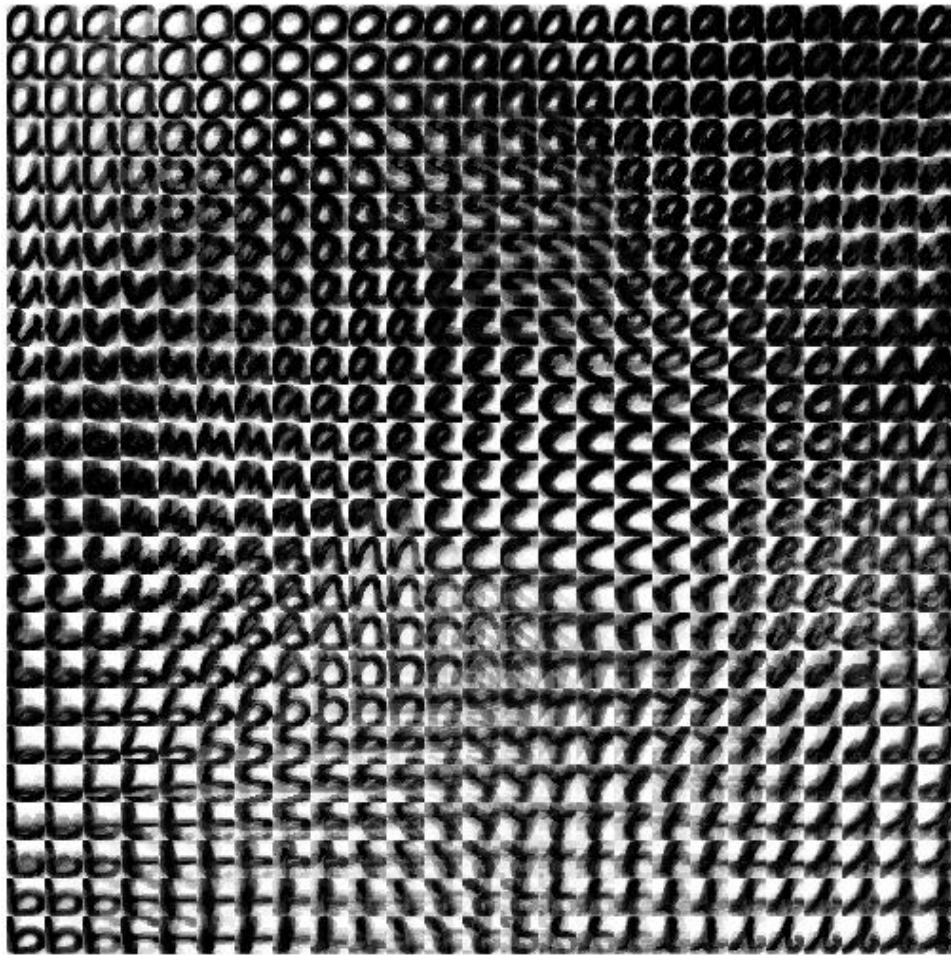


Figura 4.3 Mapa SOM generado a partir de caracteres segmentados manualmente.

Una vez entrenada la red SOM, es utilizada durante la clasificación para construir nuevos vectores de características que describan a cada segmento de palabra. La construcción de dichos vectores se hace utilizando las coordenadas de las  $k$  neuronas con mayor activación dentro del mapa para cada segmento. Se eligen  $k$  neuronas ( $k > 1$ ) debido a que existe la posibilidad de que el grupo correspondiente a la neurona ganadora no sea el correcto, el valor de  $k$  es seleccionado experimentalmente. La salida de la red SOM es un mapa ordenado, por lo que es muy probable que el segmento realmente pertenezca a un grupo cercano a la neurona ganadora en el mapa. Entonces, los nuevos vectores de características son generados a partir de la salida del mapa SOM y una métrica probabilística utilizando las

coordenadas de las  $k$  neuronas con mayor activación y la probabilidad para cada una de las  $k$  neuronas de que el segmento de entrada pertenezca al grupo al que representan. Dicha probabilidad depende de la distancia Euclidiana que tienen las neuronas con respecto a la ganadora; de esta forma la neurona ganadora tiene una mayor probabilidad, debido a su parecido con el segmento de entrada, y la probabilidad asignada a las demás neuronas disminuye conforme se alejan de la zona de mayor activación. El proceso que se lleva a cabo para asignar la probabilidad a cada segmento se explica a continuación:

Durante la fase de clasificación cada segmento  $t$  es presentado al mapa SOM, obteniendo una neurona ganadora con posición  $c_t$  en el mapa de salida, el cual representa al grupo que mejor se ajusta al segmento de entrada.

Enseguida se seleccionan las siguientes  $k-1$  neuronas  $m_{ti}$   $i = 2..k$ , con mayor activación y se designa  $m_{t1} = c_t$ . De esta forma la red SOM proporciona información sobre los  $k$  grupos más representativos para el segmento de entrada  $t$ .

El siguiente paso es calcular la probabilidad, para cada uno de los  $k$  grupos, de representar al segmento de entrada  $t$ . La probabilidad asignada es proporcional a la distancia Euclidiana entre cada neurona y la neurona ganadora; la función está definida cómo:

$$p_t(\mathbf{m}_{ti}) = \frac{\exp(-\|\mathbf{m}_{ti} - \mathbf{c}_t\|)}{\sum_{j=1}^k \exp(-\|\mathbf{m}_{tj} - \mathbf{c}_t\|)} \quad (4.1)$$

Donde  $c_t$  es un vector bidimensional definido por las coordenadas de la neurona ganadora en la red SOM al presentar el segmento  $t$ ,  $m_{ti}$  es el vector bidimensional de las coordenadas de la  $i$ -ésima neurona con mayor activación en el mapa SOM,  $i = 1..k$ , cuando el segmento  $t$  es presentado a la red SOM,  $\|\cdot\|$  es la distancia Euclidiana entre sus argumentos.

Nótese que para cada segmento  $t$ :

$$a) \sum_{i=1}^k p_t(\mathbf{m}_{ti}) = 1 \text{ y} \quad (4.2)$$

$$b) \text{ si } \|\mathbf{m}_{ti} - \mathbf{c}\| = \|\mathbf{m}_{tj} - \mathbf{c}\| \text{ entonces } p(\mathbf{m}_{ti}) = p(\mathbf{m}_{tj}). \quad (4.3)$$

La función propuesta está basada en la función de vecindad utilizada en el algoritmo de entrenamiento de la red SOM (ver sección 2.6). Dicha función es utilizada para el ajuste de los pesos provocando que la influencia de la actualización sea menor dependiendo de la distancia de las neuronas con respecto al ganador.

En resumen, el nuevo vector de características para el  $t$ -ésimo segmento está definido como:

$$\mathbf{F}_t = (c_{t1}, c_{t2}, p(c_t), m_{t21}, m_{t22}, p(\mathbf{m}_{t2}), m_{t31}, m_{t32}, p(\mathbf{m}_{t3}), \dots, m_{tk1}, m_{tk2}, p(\mathbf{m}_{tk})) \quad (4.4)$$

Nótese que existe un vector  $F_t$  para cada segmento en la palabra. Los vectores  $F_t$  son alimentados uno a uno a la red neuronal para clasificación la cual se detalla en la siguiente sección. Adicionalmente, esta forma de construir los nuevos vectores limita la variabilidad en los datos de entrada para la red que se utilizará como clasificador, debido a que las entradas estarán limitadas a los datos obtenidos del mapa SOM.

#### 4.1.4 Clasificación de palabras

La etapa de reconocimiento de palabras se lleva a cabo utilizando una red recurrente simple (SRN por sus siglas en inglés) originalmente propuesta en (Elman, 1990). Esta red permite el uso de información contextual presentada a la red, por lo tanto, puede ser utilizada para la tarea de clasificación de secuencias temporales. La red cuenta con tres capas: una capa de entrada, una capa oculta con conexiones recurrentes hacia sí misma y una capa de salida que cuenta con un nodo por cada palabra que se quiere identificar con el sistema. La Figura 4.4 muestra la arquitectura de la SRN; las líneas punteadas indican que las conexiones entre las capas tienen asociados pesos, la línea sólida indica que los enlaces no tienen asociados pesos o que



todos sus pesos son iguales a 1. Para el método de reconocimiento propuesto, los nodos de la capa de salida tienen como función de activación la función *softmax*.

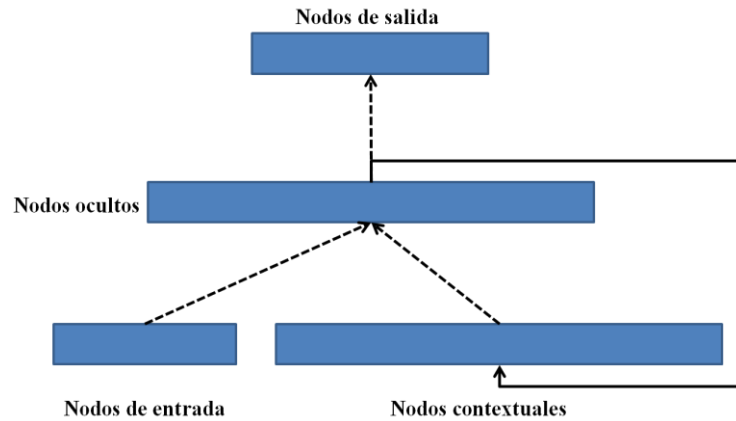


Figura 4.4 Arquitectura de una red recurrente simple (Elman, 1990).

En el método propuesto en esta investigación la entrada de esta red corresponde a los vectores  $F_t$  (ecuación 4.4) generados a partir de la red SOM. Estos vectores, que representan a segmentos que componen una palabra, son presentados a la red en forma secuencial, esto es, uno a la vez. Esta red, debido a sus conexiones recurrentes, es capaz de recordar la información contextual del pasado para poder hacer una clasificación de la secuencia de datos.

La etiqueta de palabra, para la secuencia de datos presentada a la red, corresponde a la asociada con la neurona que tiene mayor activación inmediatamente después de que se presenta el último elemento de la secuencia.

### Entrenamiento de la red

La red neuronal recurrente simple es entrenada con el algoritmo de retropropagación a través del tiempo (Werbos, 1990). Se utilizó la implementación disponible en Matlab versión 7.4.0.287. El algoritmo busca desdoblarse la red recurrente creando una red *feed-forward* y entonces utiliza el algoritmo de retropropagación estándar; el desdoble de la red se hace al presentar todos los objetos

del conjunto de entrenamiento y por cada uno de ellos se crea una nueva capa. Este tipo de entrenamiento hace que la red se ajuste a los datos como si fuera una sola serie de tiempo, sin embargo, el problema atacado en este trabajo busca reconocer varias secuencias temporales, una por cada palabra, no sólo una. Para entrenar a la SRN de forma que permita reconocer distintas secuencias el algoritmo es ejecutado de una forma incremental, en lugar de presentarle a la red todos los datos del conjunto de entrenamiento y llevar a cabo el ajuste de pesos, se presenta una secuencia de datos correspondiente a una palabra y se ajustan los pesos, este proceso se repite para cada una de las palabras del conjunto de entrenamiento.

El algoritmo de entrenamiento retropropagación a través del tiempo corresponde al paradigma supervisado por lo tanto es necesario presentar a la red un vector de entrada y su respectiva salida. La salida esperada para cada vector de entrada  $F_t$  debe ser calculada a partir del número de segmentos que contiene una palabra. Dado que la función de activación en la capa de salida es la función *softmax*, la suma de la activación de todos los nodos de la capa de salida debe ser 1 y pueden considerarse los valores de salida como probabilidades. La probabilidad esperada para el último elemento de una secuencia de datos que representa una palabra es 1 en la neurona que representa a dicha palabra. Tomando en consideración lo anterior, el valor de salida para cada elemento en la secuencia de entrada, es calculado de la siguiente forma: dada una secuencia de vectores de características  $S$  correspondiente a una palabra, con  $S = \{F_1, F_2, \dots, F_N\}$ , la salida esperada para la neurona que representa a dicha palabra es:

$$Y = \{y_1, y_2, \dots, y_N\} \text{ donde } y_i = \frac{i}{N}, i = 1 \dots N. \quad (4.6)$$

#### 4.1.5 Ejemplo del reconocimiento de palabras

A continuación se muestra un ejemplo del reconocimiento completo de la palabra “But” utilizando el método propuesto, con una red entrenada para reconocer 10 palabras distintas.

1. Se pre-procesa y segmenta la imagen. La Figura 4.5 muestra la imagen de la palabra antes y después de ser pre-procesada, las líneas marcan los puntos de segmentación generados automáticamente.

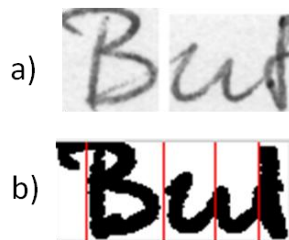


Figura 4.5 a) Imágen original de la palabra “But” obtenida de la base de datos IAM DB. b) Imágen resultante después de aplicar el algoritmo de preprocesamiento a la imagen original.

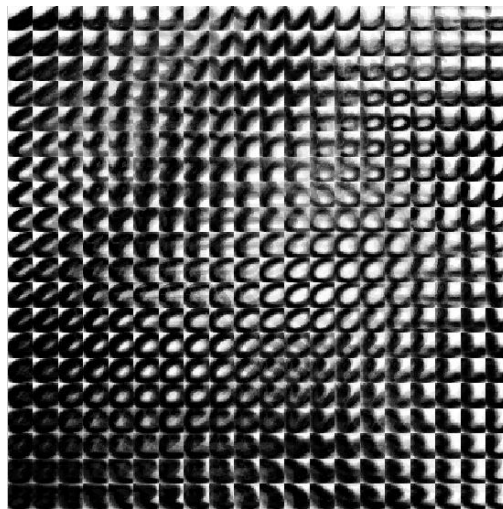


Figura 4.6 Mapa SOM entrenado a partir de segmentos de palabra generados automáticamente.

2. Los segmentos son normalizados a un tamaño de  $20 \times 20$  píxeles, se evalúan en una red SOM previamente entrenada con segmentos de palabra y se generan los nuevos vectores de entrenamiento, con  $k = 5$ . Los vectores  $F_t$  correspondientes a cada segmento de la secuencia se muestran en la Tabla 4.1.

Tabla 4.1 Vectores de características generados a partir de la red SOM de acuerdo al método propuesto.

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$
6	11	15	20	9
11	16	14	8	20
0.2092	0.2152	0.2088	0.21	0.2112
7	13	14	19	8
11	15	14	8	20
0.1987	0.1923	0.1976	0.1995	0.2007
6	12	15	19	8
12	16	13	7	19
0.1987	0.2045	0.1984	0.1955	0.1965
5	14	16	20	9
11	15	14	7	19
0.1987	0.1837	0.1968	0.1995	0.2007
7	10	15	19	7
10	16	15	9	20
0.1946	0.2043	0.1984	0.1955	0.1909
5	5	5	5	5

3. Por último, los vectores  $F_t$  son presentados a la SRN entrenada previamente para el reconocimiento de palabras, La Figura 4.7 muestra el valor de activación de los 10 nodos de la capa de salida, al presentarle a la red la secuencia de vectores  $F_t$  generados a partir de la imagen de la palabra “But”.

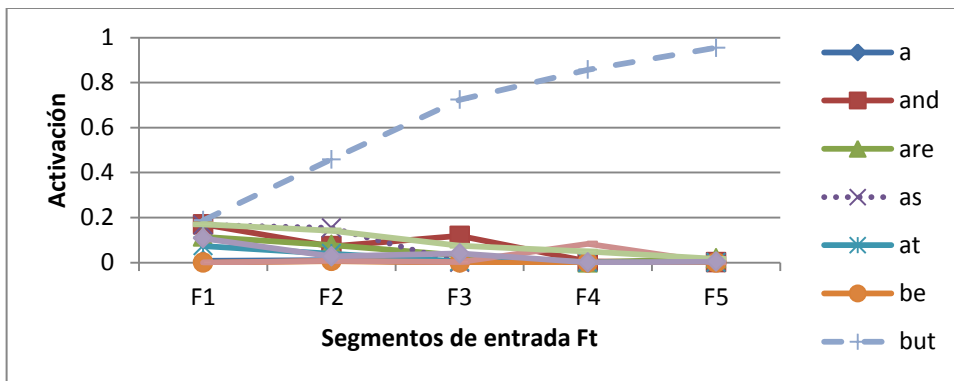
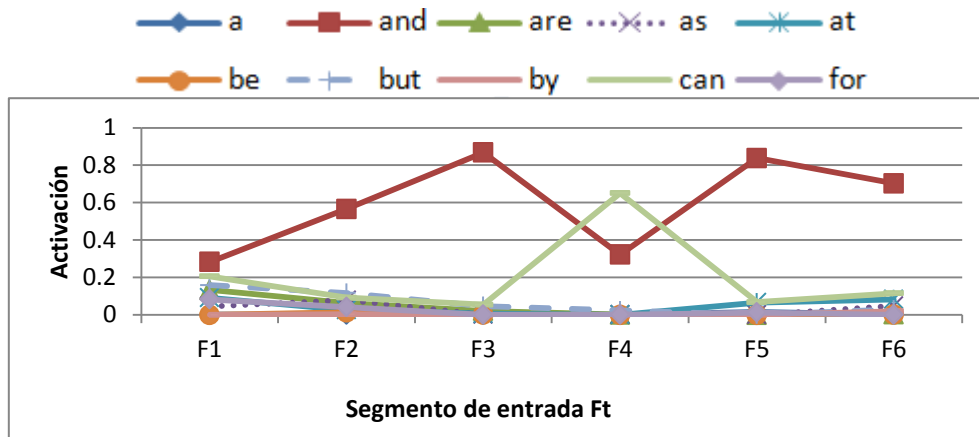
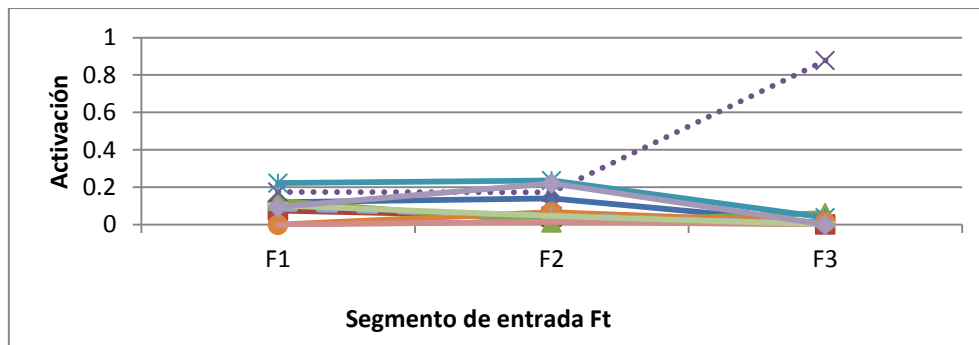


Figura 4.7 Activación de las neuronas de salida, para los 5 segmentos presentados que forman la palabra "But", los títulos de las series corresponde a la palabra representada por cada neurona en la capa de salida.

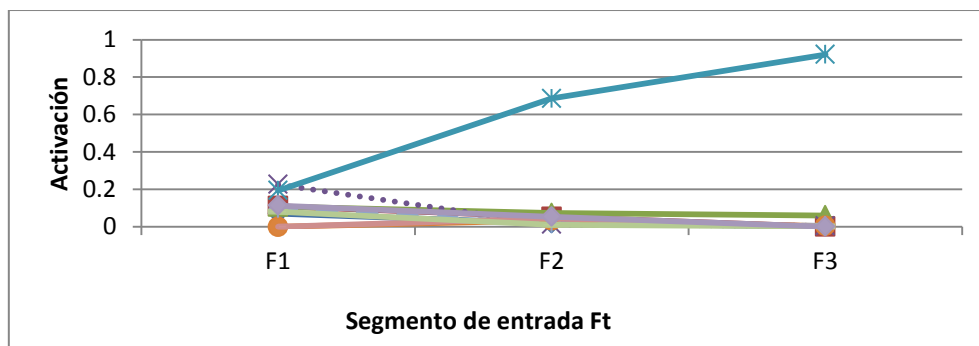
La Figura 4.8 muestra la activación en las neuronas de salida de la SRN cuando se le presentan los segmentos que representan a las palabras “and”, “as”, ”at”.



a) Activación de las neuronas de salida de la SRN cuando se le presentan los segmentos correspondientes a la palabra “and”.



b) Activación de las neuronas de salida de la SRN cuando se le presentan los segmentos correspondientes a la palabra “as”.



c) Activación de las neuronas de salida de la SRN cuando se le presentan los segmentos correspondientes a la palabra “at”.

**Figura 4.8** Activación de las neuronas de salida al presentarle a la red los vectores Ft correspondientes las palabras: a) and, b) as, c) at .



# Capítulo 5

## Experimentos y Resultados

En este capítulo se presentan los experimentos que se realizaron para probar el método propuesto, los resultados obtenidos y las medidas de evaluación utilizadas.

### 5.1 Evaluación de los resultados

Las métricas que se seleccionaron para evaluar el desempeño del método propuesto son comúnmente utilizadas en la tarea de reconocimiento de palabras y se basan en la diferencia de caracteres entre la palabra reconocida y la real. Las medidas utilizadas se describen a continuación.

#### 5.1.1 Error de reconocimiento

Es una métrica muy utilizada en las tareas de clasificación la cual corresponde al porcentaje de datos correctamente clasificados y es calculada dividiendo el número de datos incorrectamente clasificados entre el total de los datos:

$$Error = \frac{\textit{Objetos incorrectamente clasificados}}{\textit{Total de objetos}} \quad (5.1)$$

En esta métrica, se considera que un objeto está incorrectamente clasificado si la clase real del objeto no corresponde a la palabra obtenida por el clasificador, sin importar la similitud de la etiqueta asignada con la esperada.

### 5.1.2 Exactitud de palabra (*word accuracy*)

Esta métrica fue propuesta por Graves y colaboradores (2009). Dadas las etiquetas obtenidas en el proceso de reconocimiento y las etiquetas reales, para cada objeto se calcula el número mínimo de operaciones de edición para convertir la etiqueta obtenida del clasificador en la etiqueta real. Las operaciones de edición corresponden a la inserción, eliminación o sustitución de caracteres. El número total de operaciones es sumado y se divide entre el tamaño en caracteres del conjunto con el que se está probando el reconocedor. Formalmente el cálculo se define como:

$$\text{Exactitud de palabra} = 100 * \left( 1 - \frac{\text{inserciones} + \text{sustituciones} + \text{eliminaciones}}{\text{total de caracteres en el conjunto de prueba}} \right) \quad (5.2)$$

## 5.2 Descripción de la base de datos

La base de datos utilizada en este trabajo de investigación es la IAM-DB la cual está disponible para uso libre en tareas de investigación (Marti & Bunke, 1999). Dicha base de datos está compuesta por 82,000 palabras manuscritas, de las cuales existen 10,841 palabras distintas y producidas por 400 escritores distintos. De las 10,841 palabras distintas, en varios casos no se cuenta con un número adecuado de imágenes por palabra para poder entrenar el sistema propuesto en este trabajo de investigación.

Para probar el método propuesto en este trabajo de investigación se seleccionaron para los experimentos iniciales 10 palabras distintas de la base de datos IAM-DB. El conjunto de entrenamiento se formó con 200 imágenes de palabras, 20 por cada palabra distinta, mientras que para el conjunto de prueba se utilizaron 5 imágenes por cada palabra, dando un total de 50 imágenes para el conjunto de prueba. Las imágenes de cada palabra del conjunto de prueba y entrenamiento fueron escritas por distintas personas. Para realizar los experimentos, las imágenes fueron



segmentadas de dos maneras distintas: en caracteres manualmente y en segmentos de palabra buscando los puntos de segmentación automáticamente. Además, cada segmento fue normalizado a un tamaño de 20 x 20 píxeles.

La Tabla 5.1 muestra el conjunto de palabras que se utilizó para realizar estos experimentos. Las palabras fueron seleccionadas de tal forma que algunas tuvieran ciertas similitudes en el prefijo, o en alguna parte de la palabra, además se seleccionó una palabra que fuera completamente distinta a las demás. La selección de las palabras se hizo de la forma mencionada debido a que se asume que secuencias con segmentos iniciales diferentes debieran ser fácilmente distinguibles y, a su vez, secuencias con segmentos de inicio muy parecidos podrían provocar confusión. Utilizando dicho vocabulario (léxico) se buscó poder observar el comportamiento del reconocedor para secuencias en las que se pudiera confundir.

**Tabla 5.1 Vocabulario utilizado en los experimentos.**

<b>Vocabulario</b>									
a	and	are	as	at	be	but	bye	can	for

### **5.3 Experimentos realizados**

Se llevaron a cabo experimentos para dos casos distintos: segmentación manual y segmentación automática. El propósito de probar el método con segmentación manual fue observar el desempeño del método con datos de entrada que fueran menos variables que los resultantes de la segmentación automática. Con esto, pudo observarse posteriormente el efecto de la segmentación en la efectividad del método.

### 5.3.1 Experimentos con segmentación manual

Para comparar los resultados obtenidos por el método se entrenaron 3 arquitecturas de redes neuronales para la tarea de reconocimiento de palabras, con diferentes características, una de ellas corresponde al método propuesto. Todas las redes tienen 10 neuronas en la capa de salida, una por cada clase de palabra del vocabulario que se quiere reconocer. Las redes neuronales empleadas se describen a continuación:

- Red alimentada hacia adelante (identificada como FF). En esta red los datos de entrada son todos los segmentos de palabra. Los vectores de características que describen a cada segmento se construyen a partir de la concatenación de los valores de cada uno de los renglones de la imagen. Teniendo en cuenta que los segmentos de palabra son imágenes de  $20 \times 20$  píxeles; los vectores generados para cada segmento contienen 400 elementos. Dado que la red FF no posee conexiones recurrentes, no es capaz de recordar información contextual, por lo tanto, todos los segmentos de la imagen deben ser presentados al mismo tiempo. Además, aunque el número de segmentos cambia con cada palabra, la dimensión de los datos de entrada debe ser fija, por lo que la entrada depende del número máximo de segmentos que se generaron para una palabra, que para el caso del vocabulario elegido y usando segmentación manual es 3. Entonces, los vectores de características utilizados para esta red contienen 1,200 elementos ( $400 \times 3$ ).
- Red alimentada hacia adelante con datos de entrada procesados con una red SOM (identificada como SOM-FF). Los segmentos de palabra son primero procesados por una red SOM de acuerdo al método propuesto en este trabajo de investigación (ver sección 4.1.3). La creación de nuevos vectores de características se hace con un valor de  $k = 5$ , es decir, se toman en cuenta las 5 neuronas de mayor activación en el mapa de salida de la red SOM, formando, como se explica en la descripción del método propuesto (sección 4.1.3), un vector de  $3k$  elementos por segmento (ver ecuación 4.4). Debido a

que se está utilizando una red alimentada hacia adelante y por lo tanto sin memoria, los vectores que describen a todos los segmentos de una palabra deben ser presentados a la red al mismo tiempo. Entonces, los datos de entrada utilizados para esta red fueron de dimensión 45 ( $15 \times 3$ ). Nótese que este valor es mucho menor al utilizado para la red anterior (FF).

- Red recurrente simple con datos de entrada procesados por una red SOM (identificada como SOM-SRN). Esta red corresponde al método propuesto; la dimensión de los datos de entrada es  $3k = 15$  debido a que se presenta a la red solo un segmento de palabra a la vez.

Se realizaron 20 experimentos iguales para 100 configuraciones distintas de cada red. Las distintas configuraciones varían únicamente en el número de nodos en la capa oculta que tiene la red en cuestión, se probaron desde 1 hasta 100 nodos escondidos. La Tabla 5.2 muestra el error de reconocimiento obtenido para las configuraciones de red que tuvieron el mejor desempeño en cada caso. Asimismo, la Tabla 5.3 muestra la exactitud de palabra obtenida para las mismas redes. En el Apéndice E se muestran los promedios obtenidos para las 100 configuraciones.

Tabla 5.2 Error de Reconocimiento para datos segmentados manualmente.

<i>Red</i>	<i>Nodos en la capa oculta</i>	<i>Error</i>	
		<b>Conjunto de entrenamiento</b>	<b>Conjunto de Prueba</b>
FF	29	46.95% $\pm$ 27.32%	68.80% $\pm$ 14.68%
SOM-FF	11	7.93% $\pm$ 1.68%	36.80% $\pm$ 5.13%
SOM-SRN	<b>10</b>	<b>5.75%<math>\pm</math>1.34%</b>	<b>24.30%<math>\pm</math>5.12%</b>

Tabla 5.3 Exactitud de palabra para datos segmentados manualmente.

<i>Red</i>	<b>Exactitud de palabra</b>	
	<b>conjunto de entrenamiento</b>	<b>conjunto de prueba</b>
FF	55.52 $\pm$ 27.14	32.92 $\pm$ 15.51
SOM-FF	93.03 $\pm$ 1.5	66.21 $\pm$ 5.85
SOM-SRN	<b>95.42<math>\pm</math>1.21</b>	<b>78.25<math>\pm</math>3.25</b>

Puede notarse que los mejores resultados se obtienen utilizando el método propuesto en este trabajo de investigación, obteniendo un error promedio de 5.75% utilizando los datos del conjunto de entrenamiento y 24.3% utilizando los datos de prueba. Asimismo, se obtuvo una exactitud de palabra (ver ecuación 5.2) de 95.42% con los datos de entrenamiento y 78.25% con datos de prueba para la mejor configuración. Las figuras Figura 5.1 y Figura 5.2 muestran la exactitud de palabra promedio obtenida para los conjuntos de entrenamiento y prueba respectivamente, para las SRN entrenadas con distinto número de nodos en la capa oculta.

Sin embargo, aunque para el conjunto de entrenamiento el clasificador logra una exactitud de palabra promedio de 95.42%, para el conjunto de prueba se reduce hasta 78.25%. Esto puede deberse a la poca cantidad de datos que se tiene para entrenar el sistema, por lo que no se generan los suficientes modelos de los segmentos para poder cubrir más la variabilidad en los estilos de escritura.

### **5.3.2 Experimentos con segmentación automática**

Estos experimentos se llevaron a cabo para el mismo vocabulario que el utilizado en los experimentos con segmentación manual. Para este experimento, la dimensión de los datos de entrada de las redes FF y SOM-FF cambia debido que la segmentación automática produce más segmentos para una palabra que los generados manualmente. Se tomó en cuenta el número máximo de segmentos generados para una palabra, que en este caso fue 7, por lo tanto, la dimensión de los datos de entrada para la red FF es 2,800 ( $400 \times 7$ ) y para la red SOM-FF es 315 ( $45 \times 7$ ). La Tabla 5.4 muestra el error de reconocimiento promedio, obtenido utilizando el conjunto de entrenamiento, para las configuraciones de red que mostraron un mejor desempeño en cada caso. La Tabla 5.5 muestra la exactitud de palabra promedio, obtenida utilizando el conjunto de entrenamiento, para la configuración de red que mostró mejor desempeño en cada caso. Se observa que utilizando segmentación automática el desempeño es menor que utilizando segmentación manual reduciéndose, en el caso

del método propuesto, de 78.25% a 67.70% en el reconocimiento con el conjunto de prueba. La Figura 5.3 y la Figura 5.4 muestran la exactitud de palabra promedio y desviación estándar obtenidos con el conjunto de entrenamiento y prueba respectivamente, para las distintas configuraciones de las redes entrenadas.

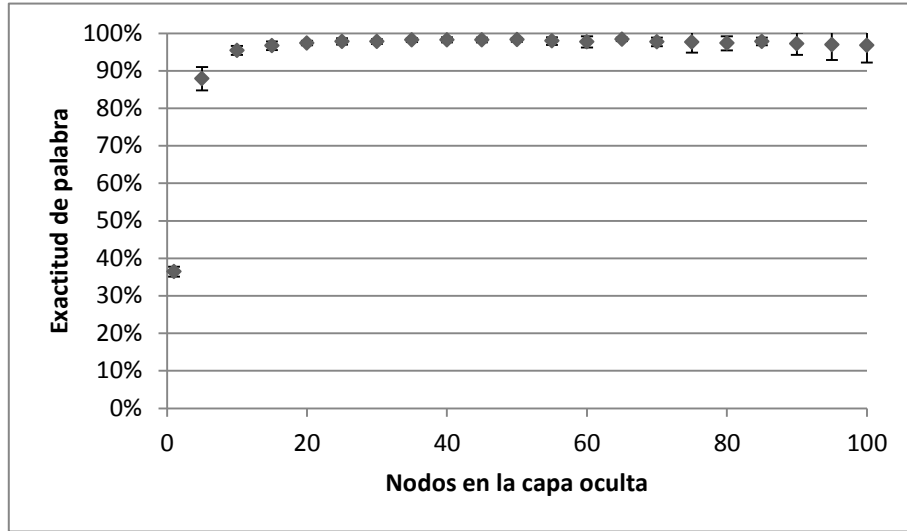


Figura 5.1 Exactitud de palabra promedio obtenido con la SRN para el conjunto de entrenamiento (segmentación manual).

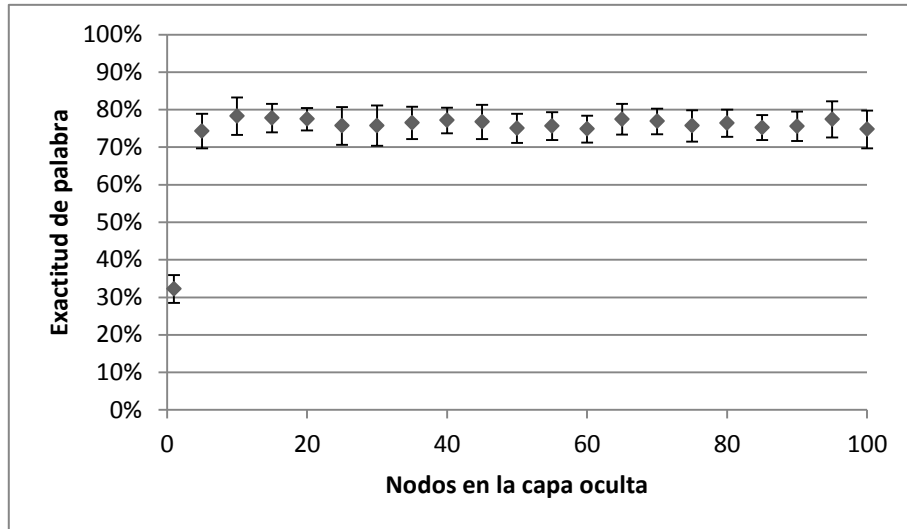


Figura 5.2 Exactitud de palabra promedio obtenido con la SRN para el conjunto de prueba (segmentación manual).

Tabla 5.4 Error de reconocimiento para 10 palabras distintas segmentadas automáticamente.

<i>Red</i>	<i>Nodos en la capa oculta para la mejor configuración</i>	<i>Error Conjunto de entrenamiento</i>	<i>Error Conjunto de Prueba</i>
FF	30	82.30% ±9.94%	86.10% ±5.78%
SOM-FF	40	12.40% ±3.65%	48.70% ±5.07%
SOM-SRN	<b>100</b>	<b>1.40%±0.42%</b>	<b>40.30%±4.69%</b>

Tabla 5.5 Exactitud de palabra obtenido con segmentación automática.

<i>Red</i>	<i>Nodos en la capa oculta</i>	<i>Exactitud de palabra</i>	
		<i>Conjunto de entrenamiento</i>	<i>Conjunto de Prueba</i>
FF	30	19.52% ±13.22%	16.75% ±10.91%
SOM-FF	40	88.34% ±3.58%	53.88% ±5.05%
SOM-SRN	<b>100</b>	<b>98.88% ±0.52%</b>	<b>67.70% ±2.58%</b>

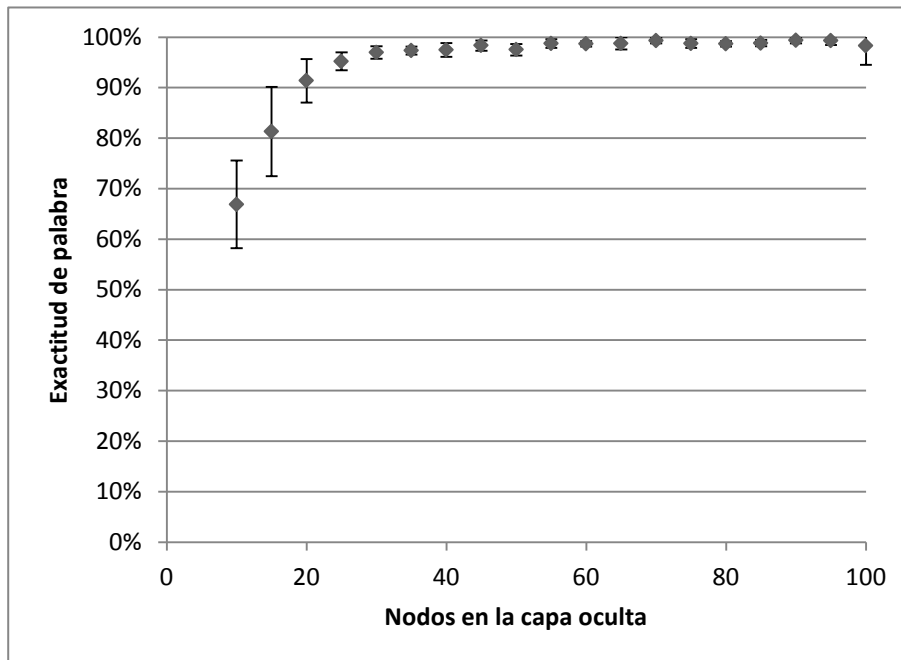
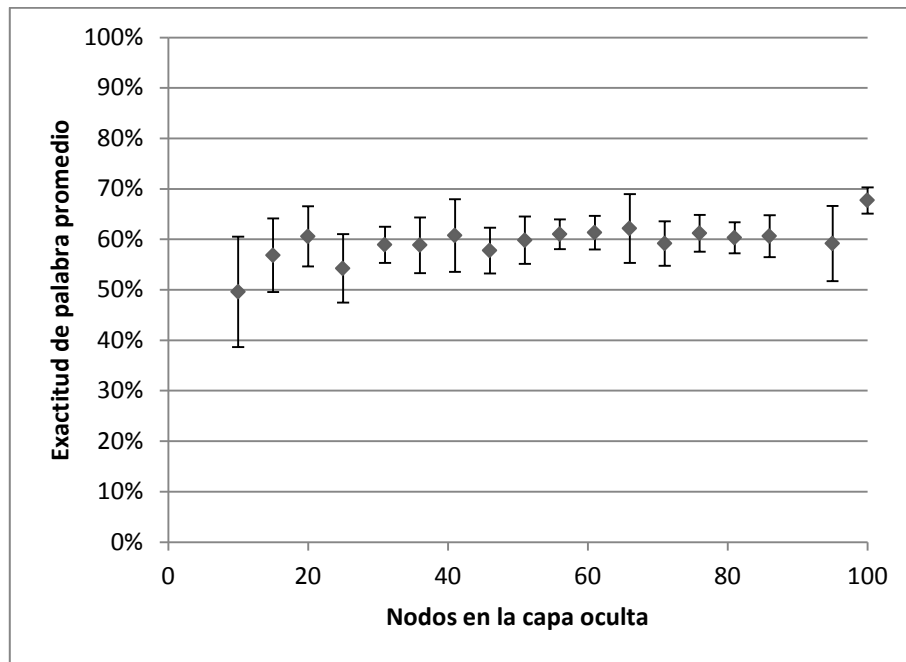


Figura 5.3 Exactitud de palabra promedio obtenido con el conjunto de entrenamiento para SRN's con distinto número de nodos en la capa oculta (segmentación automática).



**Figura 5.4** Exactitud de palabra obtenido con el conjunto de prueba para SRN's con distinto número de nodos en la capa oculta (segmentación automática).

A partir de los resultados arrojados se observa que la segmentación automática afectó el desempeño del método. Esto pudo ser generado debido a que el número de segmentos generados automáticamente varía para una misma palabra, algo que no sucede con la segmentación manual. Para tratar de contrarrestar este problema, se agregó una característica más a los vectores de entrada de la red recurrente simple; esta característica es el número de segmentos que tiene la secuencia a la que pertenece cada vector. Los resultados obtenidos al utilizar esta variación en los vectores de características se presentan en la Tabla 5.7. Nótese que el porcentaje de error se redujo en todos los casos, destacando la mejora que se logró con el método propuesto, en cuyo caso se redujo el error de 40.3% a 29.6%. La tabla completa de resultados se presenta en el Apéndice E.

Al igual que en los experimentos de segmentación automática el error al presentarle el conjunto de prueba aumenta bastante en comparación con el obtenido con el conjunto de entrenamiento y esto se puede deber a la misma causa: la poca cantidad de datos con los que se entrenó y además la variabilidad en la cantidad de

segmentos generados para una misma palabra debido al estilo de escritura, ya que, se pudo observar que para un mismo carácter se generaban una cantidad distinta de segmentos dependiendo del estilo de escritura.

**Tabla 5.6 Error de reconocimiento para 10 palabras distintas segmentadas automáticamente y agregando al vector de características el tamaño de la secuencia.**

<i>Red</i>	<i>Nodos en la capa oculta para la mejor configuración</i>	<i>Error Conjunto de entrenamiento</i>	<i>Error Conjunto de Prueba</i>
FF	20	83.30% $\pm$ 9.07%	85.90% $\pm$ 5.89%
SOM-FF	50	12.15% $\pm$ 5.04%	48.50% $\pm$ 5.10%
SOM-SRN	<b>80</b>	<b>1.65% <math>\pm</math>0.58%</b>	<b>29.60% <math>\pm</math>4.06%</b>

**Tabla 5.7 Exactitud de palabra obtenido agregando el tamaño de la secuencia al vector de características.**

<i>Red</i>	<i>Nodos en la capa oculta</i>	<i>Exactitud de palabra</i>	
		<i>Conjunto entrenamiento</i>	<i>Conjunto de Prueba</i>
FF	20	19.13% $\pm$ 10.89%	16.66% $\pm$ 9.58%
SOM-FF	50	89.05% $\pm$ 4.65%	54.96% $\pm$ 4.8%
SOM-SRN	<b>80</b>	<b>98.67% <math>\pm</math>0.85%</b>	<b>73.92% <math>\pm</math>3.09%</b>

### 5.3.3 Experimentos adicionales

Para poder observar si el utilizar más de un elemento del mapa SOM para representar a un segmento ayuda al reconocimiento, se realizaron experimento en los cuales se utilizó  $k = 1$ , de esta forma sólo se tiene información de la neurona ganadora de la red SOM para la construcción del vector de características. La Tabla 5.8 muestra el promedio del porcentaje de error y la exactitud de palabra promedio obtenidos al utilizar sólo a la neurona ganadora. Con 80 neuronas en la capa oculta se obtuvieron los mejores resultados, el cual logró una exactitud de palabra de 62.17 % el cual es menor que el 73.92% logrado utilizando  $k = 5$ . La tabla completa de resultados se presenta en el Apéndice E.



Tabla 5.8 Exactitud de palabra promedio y porcentaje de error promedio para el reconocimiento de 10 palabras con  $k=1$  y segmentación automática.

<i>Nodos en la capa oculta</i>	<i>Exactitud de palabra</i>		<i>Porcentaje de Error</i>	
	<b>Conjunto entrenamiento</b>	<b>Conjunto de Prueba</b>	<b>Conjunto entrenamiento</b>	<b>Conjunto de Prueba</b>
80	98.21% $\pm$ 1.18%	62.17% $\pm$ 4.31%	2.30% $\pm$ 1.10%	43.60% $\pm$ 3.85%

Adicionalmente, se realizaron experimentos con valores de  $k = 1..10$  para una SRN con 80 nodos en la capa oculta para observar el comportamiento del reconocimiento dependiendo del valor de  $k$ . La Figura 5.5 muestra el promedio de la exactitud de palabra en el reconocimiento de 10 palabras variando el valor de  $k$ . Donde el mejor resultado se obtiene para  $k = 5$ .

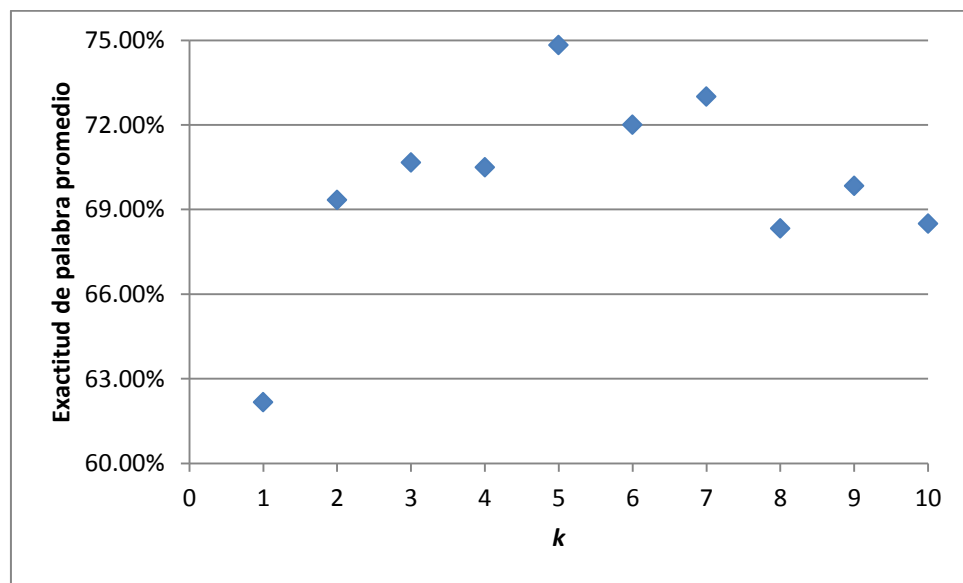


Figura 5.5 Exactitud de palabra promedio para  $k=1..10$

Para observar la mejora que se obtiene al utilizar la función de probabilidad propuesta para asignar pesos a los  $k$  neuronas con mayor activación, se llevaron a cabo experimentos sin utilizar dicha función. La Tabla 5.9 muestra el promedio del porcentaje de error y exactitud de palabra para el reconocimiento de 10 palabras utilizando el método propuesto con  $k = 5$  pero sin utilizar la función de probabilidad. El mejor resultado obtenido alcanzó un 68.00 % en la exactitud de palabra mientras

que al utilizar la función de probabilidad se obtuvo un 73.92%. La tabla completa de resultados se presenta en el Apéndice E.

**Tabla 5.9 Porcentaje de error promedio y exactitud de palabra promedio para el reconocimiento de 10 palabras utilizando el método propuesto con k=5, sin función de probabilidad y con segmentación automática.**

<i>Nodos en la capa oculta</i>	<i>Exactitud de palabra</i>		<i>Porcentaje de Error</i>	
	<b>Conjunto entrenamiento</b>	<b>Conjunto de Prueba</b>	<b>Conjunto entrenamiento</b>	<b>Conjunto de Prueba</b>
90	99.17% ±0.63%	68.00% ±6.52%	1.20% ± 0.76%	37.60% ± 6.07%

Por último la Tabla 5.10 muestra el porcentaje de error promedio para cada palabra obtenido al evaluar el método con el conjunto de entrenamiento. Se puede observar que, como se esperaba, para la palabra “for” que es completamente diferente a las otras palabras el error es el menor. Mientras que las palabras como as y at tuvieron un error muy alto y se pudo observar en los experimentos que estas palabras tendían a confundirse entre ellas. También se debe tomar en cuenta que sólo se utilizaron 5 imágenes de cada palabra para hacer los experimentos, por lo tanto el error varía muy fácilmente y por eso la desviación estándar del error es grande en la mayoría de los casos.

**Tabla 5.10 Porcentaje de Error promedio por palabra para el conjunto de prueba**

<b>Promedio del Porcentaje de Error por palabra</b>	
a	22.00% ± 6.32%
and	16.00% ± 18.38%
are	28.00% ± 8.43%
as	55.00% ± 14.14%
at	56.00% ± 10.33%
be	42.00% ± 8.43%
but	4.00% ± 6.32%
bye	32.00% ± 19.32%
can	28.00% ± 10.32%
for	<b>2.00% ± 3.00%</b>

# Capítulo 6

## Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones e ideas para trabajo futuro obtenidas a partir del trabajo de tesis presentado este documento.

### 6.1 Conclusiones

En el presente trabajo se propuso y probó un método para el reconocimiento de palabras manuscritas utilizando datos generados por distintos escritores y para un vocabulario limitado. El método se basa en una red SOM para crear modelos de segmentos de palabras de manera no supervisada y una red recurrente simple para la clasificación de secuencias compuestas de dichos segmentos.

El motivo para utilizar una red SOM para crear grupos segmentos de palabra fue probar si las relaciones de vecindad entre sus elementos podían proporcionar información útil para el clasificador. El uso de una red SOM como etapa anterior a la clasificación resultó de utilidad para lograr una mejora en el porcentaje de reconocimiento. Esto debido a que se aprovechó la relación de similitud que proporciona el mapa para poder brindar más información útil a clasificador, los resultados mostraron una mejora en el reconocimiento al utilizar regiones del mapa SOM en lugar de sólo utilizar al elemento ganador del mapa. Adicionalmente, la dimensión de los datos también se redujo ya que los vectores de entrenamiento se crearon a partir de la posición de los modelos que representan a los segmentos dentro del mapa auto-organizado, en lugar de utilizar las imágenes crudas de los segmentos.

Se propuso utilizar una función para asignar probabilidades a los  $k$  elementos utilizados del mapa SOM, de ser el modelo que representa al segmento evaluado en la red. De esta forma no tienen el mismo peso todos los elementos tomados en cuenta y en caso de utilizar un elemento poco similar tendría un peso bajo. Los resultados mostraron que el utilizar la función propuesta mejoró el reconocimiento utilizando el método propuesto, por lo que se concluye que el uso de dicha función es adecuado para evitar que datos que no aportan mucha información, tengan el mismo peso al momento de ser utilizados en el proceso de reconocimiento.

El empleo de una red recurrente simple, en lugar de redes no dinámicas, es adecuado para este tipo de tareas de clasificación en el cual se necesita información contextual del pasado y no se sabe a priori el tamaño de la secuencia de entrada. Sin embargo, la SRN es una red muy simple y el uso de otras redes especializada para el uso de información contextual podrían ayudar a mejorar el desempeño del método.

El presente trabajo de investigación aporta evidencia empírica de que el uso de una red SOM para aportar información puede ayudar a mejorar el reconocimiento de texto manuscrito. Además se diseñó un método un método de reconocimiento de palabras que toma ventaja de las características inherentes de diferentes tipos de redes neuronales, el cual obtiene buenos resultados para vocabularios reducidos, y con la ventaja de que es simple de implementar. Sin embargo, esta simplicidad implica un costo en la exactitud del método y en el tiempo utilizado para entrenar a la red recurrente simple.

## **6.2 Trabajo futuro**

Existen varias tareas involucradas en el método propuesto y en las cuales se puede continuar trabajando. El pre-procesamiento utilizado en el presente trabajo es simple; en esta tarea se podrían utilizar técnicas de preprocesamiento más complejas que permitan extraer características de la imagen que pudieran aportar más información al clasificador. El método de segmentación automática propuesta no

siempre logra obtener los mismos segmentos para la misma palabra pues como ya se mencionó en secciones anteriores; la segmentación automática de caracteres manuscritos es una tarea muy compleja y en la que se puede seguir trabajando. En el caso específico del método propuesto, se puede trabajar en un algoritmo de segmentación que busque dividir la imagen de una palabra en trazos más simples que un carácter, ya que el método reconoce una secuencia completa de elementos y no se basa en modelos de caracteres.

El entrenamiento de la SRN es supervisado, por lo tanto, requiere de muestras etiquetadas. La forma en que se etiquetaron los segmentos generados automáticamente consistió en tomar en cuenta la salida esperada para la secuencia completa y repartirla entre el número de segmentos. Sin embargo, esta no es necesariamente la forma correcta de definir las salidas esperadas para cada segmento, por lo tanto, se puede trabajar en generar un mejor etiquetado de los segmentos del conjunto de entrenamiento para lograr que la red mejore su desempeño y que el entrenamiento pudiera ser más rápido. Una idea para lograr un mejor etiquetado es utilizar información del lenguaje, de tal forma que palabras con prefijos iguales compartan la misma salida esperada, para la secuencia de entradas que corresponden a dicho prefijo.

El algoritmo utilizado para el entrenamiento de la SRN busca minimizar el error cuadrático medio entre la salida esperada y la obtenida por la red. Sin embargo, dado que lo que se busca es etiquetar palabras, podría ser conveniente proponer otra función de error que indique que tan diferente es la etiqueta de palabra que se obtuvo de la red y la etiqueta de palabra deseada. Esta función tendría que cumplir con las condiciones necesarias para ser utilizada como función objetivo en el entrenamiento de la red, con el algoritmo de retropropagación. Adicionalmente sería necesario definir una nueva forma de etiquetar los elementos del conjunto entrenamiento, para que sean adecuados a esta nueva función.

En el método propuesto la etiqueta de palabra asignada a una secuencia es la que corresponde a la neurona con mayor activación en la capa de salida de la SRN. Sin embargo, es posible que alguna otra neurona tuviera una activación similar a la ganadora, por lo tanto, se puede trabajar en un método para tomar una decisión sobre cuál es la mejor etiqueta, tomando en cuenta la activación de todas las neuronas de salida de la SRN.

Finalmente, el uso de la red SOM, para aportar más información al clasificador en la tarea de reconocimiento de palabras, puede ser utilizado en métodos ya propuestos que utilicen HMM o redes neuronales más complejas que la SRN.

## **Publicaciones derivadas del Trabajo de tesis**

- Luna Pérez, R. Gomez-Gil, P. "Unconstrained Handwritten Word Recognition Using a Combination of Neural Networks". International Conference on Soft Computing and Applications 2010, part of the World Congress of Engineering and Computer Science vol I WCECS 2010. Oct. 2010. San Francisco, USA.
- Luna Pérez, R. Gómez-Gi, P. "Redes Neuronales Auto-organizables y Recurrentes para reconocimiento temporal: el caso de palabras manuscritas" XI Encuentro de Investigación, INAOE, Nov.2010.

## Referencias

Casey, R.G. & Lecolinet, E., 1996. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(7), pp.690-706.

Chen, J.-C. & Chien, J.-T., 2009. Bayesian large margin hidden Markov models for speech recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference*, pp.3765 - 3768.

Côté, M., Lecolinet, E., Cheriet, M. & Suen, C.Y., 1998. Automatic reading of cursive scripts using a reading model and perceptual concepts. *International Journal on Document Analysis and Recognition*, pp.3-17.

Cuevas, E. & Gómez-Gil, P., 2008. PRISCUS: Reconocedor Óptico de Caracteres Manuscritos y Antiguos. *Memorias del 9o. encuentro de investigación. Instituto Nacional de Astrofísica, Óptica y Electrónica*, pp.147 - 150.

De los Santos Torres, G., 2003. *Análisis de algoritmos para reconocimiento de textos manuscritos de un solo escritor: caso Porfirio Díaz*. Tesis de Maestría en Ciencias con especialidad en Ingeniería en sistemas computacionales. Puebla: Universidad de las Américas.

Ding, Y., Ohshima, W., Kimura, F. & Shridhar, M., 2004. Local Slant Estimation for Handwritten English Words. *Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)*, pp.328 - 333.

Dong, J., Dominique, P., Krzyzak, A. & Suen, C.Y., 2005. Cursive word skew/slant corrections based on Radon transform. *Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)*, pp.478 - 483.

Elman, J.L., 1990. Finding Structure in Time. *Cognitive Science*, 14, pp.179-211.

El-Yacoubi, A., Gilloux, M., Sabourin, R. & Suen, C.Y., 1999. An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.752 - 760.

Gómez-Gil, P., De los Santos-Torres, G., Navarrete-García, J. & Ramírez-Cortés, M., 2007. The Role of Neural Networks in the Interpretation of Antique Handwritten Documents. *Hybrid Intelligent Systems*, pp.269 - 281.

Graves, A. et al., 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, pp.855 - 868.

Gupta, L., McAvoy, M. & Phegley, J., 2000. Classification of temporal sequences via prediction using the simple recurrent neural network. *Pattern Recognition*, 33, pp.1759 - 1770.

Haykin, S., 1999. *Neural Networks A Comprehensive Foundation*. 1st ed. Prentice Hall.

Kangas, J., 1994. *On the Analysis of Pattern Sequences by Self-Organizing Maps*. Tesis Doctoral. Helsinki University of Technology Laboratory of Computer and Information Science.

Kohonen, T., 1981. Automatic formation of topological maps patterns in a self-organizing system. *Proc. 2nd Scandinavian Conf. on Image Analysis*, pp.214 - 220.

Kohonen, T., 2001. *Self-Organizing Maps*. 3rd ed. Berlin Heidelberg: Springer.

Marti, U. & Bunke, H., 1999. A full English sentence database for off-line handwriting recognition. *Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pp.705 - 708, <http://www.iam.unibe.ch/fki/databases/iam-handwriting-database>.



Murillo Gil, M., 2007. *Reconocedor de textos manuscritos breves bsado en un procesameinto de palabras completas*. Tesis de Maestria. Puebla: Universidad de las Américas.

Namane, A., Guessoum, A. & Meyrueis, P., 2005. -new Holistic Handwritten Word Recognition and Its Application to French Legal Amount. *Springer-Verlag Berlin Heidelberg*, pp.654 - 663.

Plamodon, R. & Srihari, S., 2000. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(01), pp.63-84.

Rabiner, L.R. & Juang, B.H., 1986. An introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pp.5-16.

Sayre, K.M., 1973. Machine recognition of handwritten words: A project report. *Pattern Recognition*, 5(3), pp.213 - 228.

Sepp, H., Yoshua, B., Paolo, F. & Jürgen, S., 2001. Gradient Flow in Recurrent Nets: The difficulty of Learning Long-Term Dependencies. *A Field Guide to dynamical Recurrent Neural Networks*, IEEE Press.

Shapiro, L. & Stockman, G., 2001. *Computer Vision*. Prentice Hall.

Sierra Araujo, B., 2006. Introducción a las redes neuronales. In *Aprendizaje Automático: conceptos básicos y avanzados*. 1st ed. Madrid: Pearson Education. pp.101 - 131.

Toselli, A., 2004. *Reconocimiento de Texto Manuscrito Continuo*. Tesis Doctoral. Universidad Politécnica de Valencia.

Vinciarelli, A., 2002. A survey on off-line Cursive Word Recognition. *Pattern Recognition*, 35(7), pp.1433 - 1446.

Werbos, P.J., 1990. Backpropagation Through Time: What it does and How to Do It.  
*Proceedings of the IEEE*, 78(10), pp.1550 - 1560.

# Apéndice A

## Algoritmo de Retropropagación

En esta sección se presenta el algoritmo de retropropagación básico (Werbos, 1990)

Algoritmo: **NET**, Este algoritmo evalúa el patrón X en la red

Entradas: Entero m, dimensión de los datos de entrada  
Entero N, numero de nodos en la red contando los nodos de entrada  
Entero n, numero de nodos de salida  
Real X(m), patrón de entrada  
Real W(N+n,N+n), matriz de pesos

Salidas: Real x(N+n), activación de todas las neuronas  
Real Yhat(n), activación de las neuronas de salida

Paso 1: Se introducen las entradas

Para i=1..m  
 $x(i) = X(i)$

Paso 2: Se calcula la suma de las entradas de cada neurona por los pesos asociados a la sinapsis y se evalúa la función de activación

Para i=m+1..N+n  
 $net = 0$   
para j=1..i-1  
 $net = net + W(i, j) * x(j)$   
 $x(i) = 1/(1 + exp(-net))$

Paso 3: Se asignan los valores de salida

Para i=1..n  
 $Yhat(i) = x(i + N)$

Algoritmo: **F\_NET**, Se calcula el error, la deriva del error en la capa de salida, y el error que aporta cada elemento de la red para poder hacer el ajuste de los pesos.

Entradas: Real  $F\_Yhat(n)$ , derivada de la salida de la neurona  
Real  $W(N+n,N+n)$ , matriz de pesos  
Real  $x(N+n)$ , activación de todas las neuronas

Salidas:  $F\_W$

Paso 1: Se calcula la derivada de las salidas de todas las neuronas

Para  $i=1..n$

$$F\_x(i + N) = F\_Yhat(i)$$

Paso 2: Se calcula la derivada del error para cada neurona y se calcula el error hacia atrás

Para  $i=N+n..m+1, -1$

Para  $j=i+1..N+n$

$$F\_x(i) = F\_x(i) + W(j, i) * F\_net(j)$$

$$F\_net(i) = F\_x(i) * x(i) * (1 - x(i))$$

Para  $j=1..i-1$

$$F\_W(i, j) = F\_net(i) * x(j)$$

Algoritmo: **Retropropagación**

Entradas: Real  $X(T,m)$ , Conjunto de entrenamiento, con T elementos de dimensión m

Real  $Y(T,n)$ , conjunto de salidas esperadas para los datos X

Real  $W(N+n,N+n)$ , matriz de pesos

Real lr, Tasa de aprendizaje

Salidas: W

Paso 1: Se ejecuta hasta cumplir el máximo de épocas

Para época=1..MAXIMO\_EPOCAS

Paso 1.1: Para cada elemento del conjunto de entrenamiento

Para  $t=1..T$

NET( $X(t), W, x, Yhat$ )

Para  $i=1..n$

$$F\_Yhat(i) = Yhat(i) - Y(t, i)$$

F\_NET( $F\_Yhat, W, x, F\_W$ )

Paso 1.1.1:

Se actualizan los pesos

Para  $i=m+1..N+n$

Para  $j=1..i-1$

$$W(i, j) = W(i, j) - F\_W(i, j) * lr$$



## Apéndice B

### Algoritmo de Retropropagación a través del tiempo

En esta sección se presenta el algoritmo de retropropagación a través del tiempo (Werbos, 1990).

Algoritmo: **NET2**

Entradas: Entero  $m$ , dimensión de los datos de entrada  
Entero  $N$ , número de nodos en la red contando los nodos de entrada  
Entero  $n$ , número de nodos de salida  
Real  $X(m)$ , patrón de entrada  
Real  $W(N+n, N+n)$ , matriz de pesos  
Real  $W'(N+n, N+n)$ , matriz de pesos en  $t-1$   
Real  $W''(N+n, N+n)$ , matriz de pesos en  $t-2$   
Real  $x(t-1)$ , activación de todas las neuronas en  $t-1$   
Real  $x(t-2)$ , activación de todas las neuronas en  $t-2$

Salidas: Real  $x(t)$ , activación de todas las neuronas  
Real  $Yhat(n)$ , activación de las neuronas de salida

Paso 1: Se introducen las entradas

Para  $i=1..m$

$$x(i) = X(i)$$

Paso 2: Se calcula la suma de las entradas de cada neurona por los pesos asociados a la sinapsis y se evalúa la función de activación

Para  $i=m+1..N+n$

$$net = 0$$

para  $j=1..i-1$

$$net = net + W_{i,j} * x_j(t)$$

para  $j=1..N+n$   
 $net = net + W'_{i,j} * x_j(t - 1)$   
para  $j=1..N+n$   
 $net = net + W''_{i,j} * x_j(t - 2)$   
 $x_i(t) = 1/(1 + exp(-net))$

Paso 3: Se asignan los valores de salida

Para  $i=1..n$

$$Yhat(i) = x(i + N)$$

Algoritmo: **F\_NET2**

Entradas: Real F\_Yhat(n),

Real W(N+n,N+n), W'(N+n,N+n), l W''(N+n,N+n)

Real x(N+n)

Real F\_net(N+n), F\_net'(N+n), F\_net''(N+n)

Salidas: Real F\_W(N+n,N+n), F\_W'(N+n,N+n), l F\_W''(N+n,N+n)

Real F\_x(N+n)

Paso 1: Para  $i=1..N$

$$F_x(i) = 0$$

Para  $i=1..n$

$$F_x(i + N) = F_Yhat(i)$$

Paso 2: Se calcula el error hacia atras

Para  $i=N+n..m+1,-1$

Para  $j=i+1..N+n$

$$F_x(i) = F_x(i) + W(j, i) * F_net(j)$$

Para  $j=m+1..N+n$

$$F_x(i) = F_x(i) + W'(j, i) * F_net'(j) + W''(j, i) * F_net''(j)$$



$$F_{net}(i) = F_x(i) * x(i) * (1 - x(i))$$

Para  $j=1..i-1$

$$F_W(i,j) = F_W(i,j) + F_{net}(i) * x(j)$$

Para  $j=1..N+n$

$$F_W'(i,j) = F_W'(i,j) + F_{net}'(i) * x(j)$$

$$F_W''(i,j) = F_W''(i,j) + F_{net}''(i) * x(j)$$

Algoritmo: **Retropropagación a través del tiempo**

Entradas: Real  $X(T,m)$ , Conjunto de entrenamiento, con  $T$  elementos de dimensión  $m$

Real  $Y(T,n)$ , conjunto de salidas esperadas para los datos  $X$

Real  $W(N+n,N+n)$ , matriz de pesos

Real  $lr$ , Tasa de aprendizaje

Salidas:  $W$

Paso 1: Se ejecuta hasta cumplir el máximo de épocas

Para época=1..MAXIMO\_EPOCAS

Paso 1.1: Para cada elemento del conjunto de entrenamiento

Para  $t=1..T$

NET2( $X(t), W, W', W'', x(t-2), x(t-1), x(t), Yhat(t)$ )

Para  $i=m+1..N+n$

$$F_{net}'(i) = 0$$

$$F_{net}''(i) = 0$$

Para  $j=1..N+n$

$$F_w(i,j) = 0$$

$$F_w'(i,j) = 0$$

$$F_w''(i,j) = 0$$

Para  $t=T, .1$

Para  $i=1..n$

$$F\_Yhat(i) = Yhat(t, i) - Y(t, i)$$

F\_NET2(F\_Yhat,W,W',W'',x(t),F\_net,

$$F\_net',F\_net'',F\_W,F\_W',F\_W'')$$

Para  $i=m+1..N+n$

$$F\_net''(t) = F\_net'(t)$$

$$F\_net'(t) = F\_net(t)$$

Para  $i=m+1..N+n$

Para  $j=m+1..N+n$

$$W(i, j) = W(i, j) - lr * F\_W(i, j)$$

$$W'(i, j) = W'(i, j) - lr * F\_W'(i, j)$$

$$W''(i, j) = W''(i, j) - lr * F\_W''(i, j)$$

# Apéndice C

## Algoritmo de Preprocesamiento

En esta sección se presentan los algoritmos que componen el preprocesamiento de imágenes empleado en el presente trabajo.

Algoritmo: **Umbral**

Entradas: Imagen RGB, imagen

Número, umbral

Salidas: Imagen Binaria, imagen\_binaria

Paso 1: Para cada pixel  $p_{xy}$  en imagen  
Si  $p_{xy} > \text{umbral}$   
    imagen\_binaria( $x, y$ )  $\leftarrow$  0  
en caso contrario  
    imagen\_binaria( $x, y$ )  $\leftarrow$  1

Algoritmo: **Transformación Empuje**

Entrada: Imagen Binaria  $img$

Real  $sx$ , tangente del ángulo de inclinación

Salida: Imagen Binaria  $img2$

Paso 1: Para cada pixel  $img_{x,y}$  en imagen  
Si  $img_{x,y}$  es negro  
     $x2 = x + sx * y$   
     $img2_{x2,y} = img_{x,y}$

Algoritmo: **Corrección de Inclinación** (Ding et al., 2004)

Entrada: ImagenBinaria Img

Sailida: Img

Paso 1: Aplicar seguimiento de borde a la imagen de entrada para obtener el código cadena.

Paso 2: Estimar la inclinación loca utilizando la distribución de frecuencia acumulada de el código cadena

Paso 3: Aplicar la transformación de empuje para corregir la inclinación.

## Apéndice D

### Algoritmo de Segmentación

En esta sección se presenta el algoritmo para obtener los puntos de segmentación en una imagen que contiene una palabra. Dicho algoritmo busca los cambios de dirección en el borde inferior de la imagen, cuando el recorrido del borde inferior cambia de ascender a descender se toma ese punto como un punto de segmentación.

#### Algoritmo: **Obtener Puntos de Segmentación**

Entradas: Imagen, imagen

Salidas: Lista, puntos\_segmentacion

Paso 1: Para cada columna x de imagen

$\text{minimo}[x] \leftarrow$  numero de fila del pixel negro inferior de la columna x

Paso 2: Para x=0 hasta tamaño de minimo

Mientras los valores  $\text{minimo}[x]$  decrezcan o sean iguales

$x \leftarrow x+1$

Se agrega a puntos\_segmentacion el punto x

Mientras los valores  $\text{min}[x]$  aumenten o sean iguales

$x \leftarrow x+1$



# Apéndice E

## Tablas Detalladas

En esta sección se muestran las tablas detalladas con los promedios de los resultados obtenidos en los experimentos realizados.

**Tabla E.0.1 Porcentaje de error promedio y exactitud de palabra con sus respectivas desviaciones estándar obtenidos variando el número de nodos en la capa oculta de la red FF para el reconocimiento de palabras. Segmentación manual.**

Nodos	Error Conjunto entrenamiento	Error Conjunto de Prueba	Exactitud de Palabra Conjunto de entrenamiento	Exactitud de Palabra Conjunto de Prueba
1	85.90% ± 8.51%	88.00% ± 4.15%	16.30% ± 14.80%	14.33% ± 12.41%
2	75.03% ± 15.23%	82.20% ± 8.73%	26.07% ± 18.70%	18.96% ± 14.03%
3	82.48% ± 15.95%	86.80% ± 6.53%	16.39% ± 22.48%	12.71% ± 16.77%
4	78.55% ± 14.99%	84.40% ± 7.07%	16.32% ± 23.07%	11.21% ± 17.24%
5	69.10% ± 20.00%	80.50% ± 10.36%	35.28% ± 21.27%	24.50% ± 14.49%
6	76.93% ± 13.59%	84.40% ± 6.92%	21.80% ± 17.57%	14.79% ± 13.90%
7	64.20% ± 21.56%	78.30% ± 10.67%	35.58% ± 24.76%	22.29% ± 14.66%
8	70.68% ± 23.75%	80.00% ± 12.95%	32.06% ± 24.52%	23.88% ± 16.48%
9	66.63% ± 17.93%	78.30% ± 9.87%	33.30% ± 22.33%	22.71% ± 15.23%
10	73.35% ± 17.34%	81.20% ± 9.34%	30.15% ± 21.39%	23.17% ± 15.84%
11	64.85% ± 25.21%	78.40% ± 13.60%	34.77% ± 29.03%	21.42% ± 20.00%
12	58.13% ± 22.06%	74.80% ± 10.33%	44.07% ± 24.05%	28.38% ± 14.07%
13	64.83% ± 25.60%	79.10% ± 11.69%	34.56% ± 29.59%	21.50% ± 18.40%
14	71.75% ± 20.65%	81.40% ± 8.80%	30.33% ± 22.47%	20.58% ± 13.31%
15	59.60% ± 22.18%	76.40% ± 11.89%	41.90% ± 23.46%	26.42% ± 14.27%
16	58.48% ± 24.51%	75.00% ± 12.23%	44.68% ± 23.99%	29.54% ± 14.36%
17	60.45% ± 23.58%	75.20% ± 13.30%	42.78% ± 23.24%	29.33% ± 14.18%
18	59.23% ± 27.34%	74.90% ± 14.07%	43.13% ± 25.87%	29.08% ± 14.48%
19	52.30% ± 30.65%	73.90% ± 14.59%	46.05% ± 34.06%	26.33% ± 20.47%
20	49.43% ± 28.03%	70.60% ± 15.21%	53.05% ± 27.42%	33.12% ± 17.25%
21	69.38% ± 22.26%	80.10% ± 11.29%	32.93% ± 25.19%	22.83% ± 16.01%
22	64.65% ± 23.51%	77.90% ± 12.87%	35.28% ± 26.97%	22.79% ± 19.45%
23	58.48% ± 20.28%	74.30% ± 11.13%	41.48% ± 24.96%	27.33% ± 17.32%

Nodos	Error Conjunto entrenamiento	Error Conjunto de Prueba	Exactitud de Palabra Conjunto de entrenamiento	Exactitud de Palabra Conjunto de Prueba
24	63.18% ± 23.73%	78.80% ± 11.11%	42.03% ± 20.98%	26.88% ± 8.92%
25	63.25% ± 21.33%	77.90% ± 10.10%	38.42% ± 22.87%	25.21% ± 13.42%
26	74.60% ± 19.83%	83.20% ± 9.28%	25.79% ± 21.83%	17.75% ± 13.83%
27	61.63% ± 30.90%	76.40% ± 15.16%	39.01% ± 33.65%	25.54% ± 20.58%
28	60.50% ± 28.11%	77.80% ± 12.70%	36.25% ± 34.35%	20.21% ± 20.25%
<b>29</b>	<b>46.95% ± 27.32%</b>	<b>69.80% ± 14.65%</b>	<b>55.52% ± 27.14%</b>	<b>32.92% ± 15.55%</b>
30	68.15% ± 21.40%	80.20% ± 9.73%	31.52% ± 23.84%	20.42% ± 14.71%
31	63.95% ± 28.72%	79.30% ± 12.18%	39.48% ± 27.76%	24.83% ± 12.95%
32	63.65% ± 27.54%	78.50% ± 13.30%	38.32% ± 27.56%	24.58% ± 15.54%
33	71.00% ± 24.57%	80.20% ± 13.52%	31.83% ± 27.95%	22.29% ± 17.62%
34	66.45% ± 29.64%	79.40% ± 13.33%	34.46% ± 31.10%	22.08% ± 16.80%
35	73.38% ± 24.53%	82.10% ± 12.39%	33.33% ± 24.15%	25.17% ± 13.53%
36	72.78% ± 27.59%	81.70% ± 13.86%	26.93% ± 30.40%	18.96% ± 18.56%
37	60.73% ± 33.96%	76.60% ± 15.57%	40.10% ± 36.52%	26.04% ± 21.33%
38	69.98% ± 27.60%	80.00% ± 15.25%	33.99% ± 27.73%	23.88% ± 18.16%
39	60.70% ± 32.38%	76.30% ± 16.24%	42.16% ± 32.43%	28.13% ± 18.91%
40	84.38% ± 17.37%	86.60% ± 10.49%	19.17% ± 20.60%	17.08% ± 15.75%
41	59.65% ± 32.61%	75.20% ± 16.66%	43.43% ± 32.40%	28.37% ± 17.42%
42	67.48% ± 33.78%	80.20% ± 15.05%	36.64% ± 33.60%	24.46% ± 17.48%
43	57.35% ± 31.09%	73.70% ± 16.97%	41.61% ± 34.82%	26.33% ± 22.08%
44	65.80% ± 32.37%	78.10% ± 16.11%	35.89% ± 32.86%	25.21% ± 18.82%
45	73.65% ± 25.22%	82.90% ± 12.44%	30.15% ± 24.75%	22.12% ± 15.19%
46	76.43% ± 28.19%	84.20% ± 12.16%	27.49% ± 29.99%	20.67% ± 17.72%
47	84.55% ± 18.14%	86.50% ± 12.41%	23.48% ± 20.42%	21.50% ± 16.22%
48	65.80% ± 35.34%	78.40% ± 16.90%	40.94% ± 32.60%	29.58% ± 16.98%
49	79.78% ± 25.46%	85.00% ± 12.46%	20.18% ± 28.79%	15.04% ± 17.81%
50	65.85% ± 31.59%	77.90% ± 15.90%	36.00% ± 33.25%	24.88% ± 19.47%
51	71.53% ± 31.15%	80.50% ± 16.54%	34.26% ± 29.51%	26.21% ± 17.92%
52	77.75% ± 26.20%	84.00% ± 13.25%	23.37% ± 27.16%	18.29% ± 17.76%
53	75.05% ± 27.58%	83.20% ± 13.26%	27.69% ± 30.54%	20.46% ± 19.18%
54	85.80% ± 18.78%	87.80% ± 9.84%	17.07% ± 22.42%	15.17% ± 16.12%
55	82.88% ± 22.11%	87.60% ± 7.42%	16.90% ± 25.95%	12.50% ± 15.40%
56	76.35% ± 28.71%	84.90% ± 11.08%	23.36% ± 32.43%	15.21% ± 18.71%
57	79.35% ± 26.03%	84.30% ± 14.47%	21.86% ± 29.02%	17.04% ± 19.53%
58	82.25% ± 23.99%	86.50% ± 11.01%	16.99% ± 27.02%	13.04% ± 16.78%
59	76.28% ± 28.90%	83.70% ± 14.79%	22.01% ± 32.42%	15.96% ± 21.87%
60	78.90% ± 27.23%	85.80% ± 11.11%	22.86% ± 28.95%	16.58% ± 16.57%
61	82.35% ± 23.56%	86.60% ± 10.64%	16.51% ± 26.84%	12.63% ± 16.80%
62	90.00% ± 0.00%	90.00% ± 0.00%	13.75% ± 12.17%	13.75% ± 12.17%



<b>Nodos</b>	<b>Error Conjunto entrenamiento</b>	<b>Error Conjunto de Prueba</b>	<b>Exactitud de Palabra Conjunto de entrenamiento</b>	<b>Exactitud de Palabra Conjunto de Prueba</b>
63	82.73% ± 22.50%	86.40% ± 11.55%	21.29% ± 26.03%	18.00% ± 18.56%
64	83.18% ± 21.20%	87.70% ± 7.43%	18.39% ± 24.77%	14.21% ± 14.83%
65	90.00% ± 0.00%	90.00% ± 0.00%	11.04% ± 11.01%	11.04% ± 11.01%
66	85.68% ± 19.34%	87.50% ± 11.18%	13.19% ± 21.68%	11.42% ± 14.84%
67	84.10% ± 19.28%	87.60% ± 7.50%	16.51% ± 21.78%	13.46% ± 14.10%
68	80.48% ± 24.36%	85.20% ± 12.37%	14.54% ± 28.30%	10.25% ± 18.38%
69	90.00% ± 0.00%	90.00% ± 0.00%	6.04% ± 13.75%	6.04% ± 13.75%
70	77.98% ± 29.41%	83.50% ± 15.88%	27.64% ± 29.49%	22.58% ± 18.19%
71	90.00% ± 0.00%	90.00% ± 0.00%	13.33% ± 12.21%	13.33% ± 12.21%
72	86.75% ± 14.53%	89.00% ± 4.47%	11.63% ± 19.93%	9.50% ± 13.75%
73	90.00% ± 0.00%	90.00% ± 0.00%	15.42% ± 13.04%	15.42% ± 13.04%
74	86.13% ± 17.33%	88.10% ± 8.50%	16.03% ± 20.74%	14.33% ± 15.27%
75	90.00% ± 0.00%	90.00% ± 0.00%	17.92% ± 11.07%	17.92% ± 11.07%
76	90.00% ± 0.00%	90.00% ± 0.00%	9.79% ± 8.14%	9.79% ± 8.14%
77	90.00% ± 0.00%	90.00% ± 0.00%	7.29% ± 13.64%	7.29% ± 13.64%
78	86.05% ± 17.66%	88.50% ± 6.71%	16.01% ± 21.74%	13.71% ± 14.66%
79	86.53% ± 15.54%	87.60% ± 10.73%	15.41% ± 19.18%	14.29% ± 15.29%
80	85.60% ± 19.68%	88.10% ± 8.50%	19.94% ± 21.75%	17.58% ± 13.90%
81	90.00% ± 0.00%	90.00% ± 0.00%	12.50% ± 10.81%	12.50% ± 10.81%
82	90.00% ± 0.00%	90.00% ± 0.00%	9.58% ± 13.92%	9.58% ± 13.92%
83	90.00% ± 0.00%	90.00% ± 0.00%	14.38% ± 11.82%	14.38% ± 11.82%
84	90.00% ± 0.00%	90.00% ± 0.00%	9.79% ± 15.43%	9.79% ± 15.43%
85	90.00% ± 0.00%	90.00% ± 0.00%	13.75% ± 10.91%	13.75% ± 10.91%
86	90.00% ± 0.00%	90.00% ± 0.00%	13.54% ± 12.16%	13.54% ± 12.16%
87	90.00% ± 0.00%	90.00% ± 0.00%	8.33% ± 10.30%	8.33% ± 10.30%
88	90.00% ± 0.00%	90.00% ± 0.00%	11.67% ± 13.83%	11.67% ± 13.83%
89	90.00% ± 0.00%	90.00% ± 0.00%	12.71% ± 12.35%	12.71% ± 12.35%
90	85.63% ± 19.57%	87.90% ± 9.39%	16.99% ± 22.13%	14.92% ± 14.89%
91	85.63% ± 19.57%	87.70% ± 10.29%	16.78% ± 22.44%	14.92% ± 15.93%
92	90.00% ± 0.00%	90.00% ± 0.00%	14.17% ± 11.82%	14.17% ± 11.82%
93	90.00% ± 0.00%	90.00% ± 0.00%	8.54% ± 10.07%	8.54% ± 10.07%
94	90.00% ± 0.00%	90.00% ± 0.00%	11.25% ± 13.11%	11.25% ± 13.11%
95	85.60% ± 19.68%	87.40% ± 11.63%	12.63% ± 24.82%	11.04% ± 19.51%
96	90.00% ± 0.00%	90.00% ± 0.00%	13.75% ± 11.95%	13.75% ± 11.95%
97	90.00% ± 0.00%	90.00% ± 0.00%	15.00% ± 10.59%	15.00% ± 10.59%
98	85.50% ± 20.12%	87.30% ± 12.07%	17.50% ± 22.64%	15.67% ± 16.17%
99	86.03% ± 17.78%	88.10% ± 8.50%	14.55% ± 21.74%	12.54% ± 15.16%
100	90.00% ± 0.00%	90.00% ± 0.00%	7.92% ± 14.36%	7.92% ± 14.36%
101	90.00% ± 0.00%	90.00% ± 0.00%	14.79% ± 13.41%	14.79% ± 13.41%

<b>Nodos</b>	<b>Error Conjunto entrenamiento</b>		<b>Error Conjunto de Prueba</b>		<b>Exactitud de Palabra Conjunto de entrenamiento</b>		<b>Exactitud de Palabra Conjunto de Prueba</b>	
102	90.00%	± 0.00%	90.00%	± 0.00%	13.75%	± 10.65%	13.75%	± 10.65%
103	90.00%	± 0.00%	90.00%	± 0.00%	14.58%	± 12.28%	14.58%	± 12.28%
104	90.00%	± 0.00%	90.00%	± 0.00%	10.00%	± 12.64%	10.00%	± 12.64%
105	90.00%	± 0.00%	90.00%	± 0.00%	11.04%	± 11.65%	11.04%	± 11.65%
106	90.00%	± 0.00%	90.00%	± 0.00%	9.58%	± 13.66%	9.58%	± 13.66%
107	90.00%	± 0.00%	90.00%	± 0.00%	7.71%	± 14.20%	7.71%	± 14.20%
108	90.00%	± 0.00%	90.00%	± 0.00%	12.92%	± 12.23%	12.92%	± 12.23%
109	90.00%	± 0.00%	90.00%	± 0.00%	9.38%	± 12.89%	9.38%	± 12.89%
110	90.00%	± 0.00%	90.00%	± 0.00%	13.33%	± 12.80%	13.33%	± 12.80%
111	90.00%	± 0.00%	90.00%	± 0.00%	10.42%	± 8.81%	10.42%	± 8.81%
112	90.00%	± 0.00%	90.00%	± 0.00%	9.17%	± 12.80%	9.17%	± 12.80%
113	90.00%	± 0.00%	90.00%	± 0.00%	12.71%	± 11.51%	12.71%	± 11.51%
114	86.18%	± 17.11%	87.90%	± 9.39%	17.45%	± 20.65%	15.96%	± 15.95%
115	90.00%	± 0.00%	90.00%	± 0.00%	10.21%	± 12.28%	10.21%	± 12.28%
116	90.00%	± 0.00%	90.00%	± 0.00%	9.58%	± 13.39%	9.58%	± 13.39%
117	90.00%	± 0.00%	90.00%	± 0.00%	6.04%	± 11.74%	6.04%	± 11.74%
118	90.00%	± 0.00%	90.00%	± 0.00%	9.17%	± 15.33%	9.17%	± 15.33%
119	90.00%	± 0.00%	90.00%	± 0.00%	6.87%	± 11.96%	6.87%	± 11.96%
120	90.00%	± 0.00%	90.00%	± 0.00%	12.29%	± 12.13%	12.29%	± 12.13%
121	90.00%	± 0.00%	90.00%	± 0.00%	13.33%	± 11.68%	13.33%	± 11.68%
122	90.00%	± 0.00%	90.00%	± 0.00%	11.25%	± 13.73%	11.25%	± 13.73%
123	90.00%	± 0.00%	90.00%	± 0.00%	6.46%	± 12.35%	6.46%	± 12.35%
124	90.00%	± 0.00%	90.00%	± 0.00%	11.25%	± 11.40%	11.25%	± 11.40%
125	90.00%	± 0.00%	90.00%	± 0.00%	11.46%	± 13.31%	11.46%	± 13.31%
126	90.00%	± 0.00%	90.00%	± 0.00%	8.54%	± 14.14%	8.54%	± 14.14%
127	90.00%	± 0.00%	90.00%	± 0.00%	13.33%	± 15.33%	13.33%	± 15.33%
128	90.00%	± 0.00%	90.00%	± 0.00%	6.46%	± 10.94%	6.46%	± 10.94%
129	90.00%	± 0.00%	90.00%	± 0.00%	14.58%	± 12.72%	14.58%	± 12.72%
130	90.00%	± 0.00%	90.00%	± 0.00%	11.67%	± 11.03%	11.67%	± 11.03%
131	90.00%	± 0.00%	90.00%	± 0.00%	7.71%	± 11.88%	7.71%	± 11.88%
132	90.00%	± 0.00%	90.00%	± 0.00%	14.79%	± 14.08%	14.79%	± 14.08%
133	90.00%	± 0.00%	90.00%	± 0.00%	13.13%	± 11.57%	13.13%	± 11.57%
134	90.00%	± 0.00%	90.00%	± 0.00%	11.04%	± 14.39%	11.04%	± 14.39%
135	90.00%	± 0.00%	90.00%	± 0.00%	10.63%	± 8.39%	10.63%	± 8.39%
136	90.00%	± 0.00%	90.00%	± 0.00%	15.83%	± 11.28%	15.83%	± 11.28%
137	90.00%	± 0.00%	90.00%	± 0.00%	8.96%	± 15.13%	8.96%	± 15.13%
138	90.00%	± 0.00%	90.00%	± 0.00%	8.75%	± 11.06%	8.75%	± 11.06%
139	90.00%	± 0.00%	90.00%	± 0.00%	13.13%	± 12.56%	13.13%	± 12.56%
140	90.00%	± 0.00%	90.00%	± 0.00%	10.63%	± 12.28%	10.63%	± 12.28%

Nodos	Error Conjunto entrenamiento	Error Conjunto de Prueba	Exactitud de Palabra Conjunto de entrenamiento	Exactitud de Palabra Conjunto de Prueba
141	90.00% ± 0.00%	90.00% ± 0.00%	15.42% ± 11.87%	15.42% ± 11.87%
142	90.00% ± 0.00%	90.00% ± 0.00%	13.13% ± 11.96%	13.13% ± 11.96%
143	90.00% ± 0.00%	90.00% ± 0.00%	14.58% ± 11.35%	14.58% ± 11.35%
144	86.48% ± 15.76%	88.80% ± 5.37%	13.06% ± 19.65%	11.04% ± 13.20%
145	90.00% ± 0.00%	90.00% ± 0.00%	8.13% ± 15.26%	8.13% ± 15.26%
146	90.00% ± 0.00%	90.00% ± 0.00%	10.42% ± 13.82%	10.42% ± 13.82%
147	90.00% ± 0.00%	90.00% ± 0.00%	15.83% ± 13.63%	15.83% ± 13.63%
148	90.00% ± 0.00%	90.00% ± 0.00%	12.71% ± 10.16%	12.71% ± 10.16%
149	90.00% ± 0.00%	90.00% ± 0.00%	7.71% ± 15.13%	7.71% ± 15.13%
150	90.00% ± 0.00%	90.00% ± 0.00%	12.29% ± 14.14%	12.29% ± 14.14%

Tabla E.0.2 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red SOM-FF con k=5 y segmentación manual.

Nodos	Error Conjunto entrenamiento	Error Conjunto de Prueba	Exactitud de Palabra Conjunto de entrenamiento	Exactitud de Palabra Conjunto de Prueba
1	62.03% ± 7.40%	71.80% ± 7.40%	39.72% ± 10.14%	30.71% ± 10.26%
2	29.75% ± 6.90%	52.10% ± 8.98%	72.20% ± 5.40%	52.04% ± 9.73%
3	16.18% ± 1.98%	40.40% ± 5.53%	85.66% ± 2.43%	62.79% ± 4.33%
4	13.45% ± 2.33%	38.70% ± 6.72%	88.01% ± 2.14%	63.92% ± 7.17%
5	11.98% ± 2.21%	37.30% ± 4.37%	89.25% ± 2.21%	65.25% ± 4.04%
6	11.00% ± 1.75%	38.80% ± 5.82%	90.20% ± 1.88%	64.54% ± 5.65%
7	9.63% ± 1.57%	36.90% ± 5.05%	91.58% ± 1.53%	65.33% ± 5.00%
8	8.68% ± 1.12%	37.20% ± 4.96%	92.36% ± 1.04%	66.33% ± 5.59%
9	8.58% ± 1.44%	37.60% ± 6.82%	92.58% ± 1.45%	65.08% ± 5.89%
10	9.23% ± 2.19%	39.30% ± 4.87%	91.85% ± 2.25%	64.29% ± 3.98%
<b>11</b>	<b>7.93% ± 1.68%</b>	<b>36.80% ± 5.13%</b>	<b>93.03% ± 1.50%</b>	<b>66.21% ± 5.85%</b>
12	8.08% ± 2.03%	38.20% ± 5.39%	92.92% ± 1.95%	64.29% ± 5.63%
13	6.75% ± 1.55%	39.10% ± 5.25%	94.26% ± 1.48%	63.17% ± 4.43%
14	7.83% ± 1.58%	38.50% ± 4.20%	92.93% ± 1.61%	63.17% ± 4.17%
15	8.13% ± 1.38%	37.20% ± 5.33%	93.05% ± 1.36%	65.25% ± 4.83%
16	7.78% ± 2.35%	37.40% ± 5.92%	93.08% ± 2.41%	64.75% ± 6.12%
17	7.78% ± 1.19%	37.00% ± 4.83%	93.33% ± 1.25%	65.63% ± 3.99%
18	7.23% ± 2.57%	39.30% ± 3.51%	93.77% ± 2.39%	63.33% ± 3.57%
19	6.85% ± 1.86%	38.80% ± 5.37%	94.16% ± 1.75%	63.04% ± 5.60%
20	7.23% ± 2.35%	39.10% ± 4.92%	93.69% ± 1.89%	63.50% ± 5.53%
21	6.85% ± 1.42%	36.60% ± 4.55%	94.03% ± 1.52%	65.67% ± 3.92%
22	6.55% ± 2.76%	37.40% ± 4.99%	94.50% ± 2.19%	65.42% ± 4.76%
23	6.95% ± 2.67%	39.50% ± 5.19%	94.03% ± 2.14%	63.21% ± 4.87%
24	5.93% ± 1.89%	39.20% ± 5.78%	94.83% ± 1.83%	61.92% ± 5.35%
25	6.48% ± 1.66%	35.80% ± 4.35%	94.42% ± 1.41%	64.83% ± 3.76%
26	5.95% ± 1.44%	35.20% ± 4.83%	94.92% ± 1.44%	65.88% ± 4.46%
27	6.15% ± 2.46%	37.00% ± 4.47%	94.74% ± 2.14%	65.83% ± 4.11%

<b>Nodos</b>	<b>Error Conjunto entrenamiento</b>	<b>Error Conjunto de Prueba</b>	<b>Exactitud de Palabra Conjunto de entrenamiento</b>	<b>Exactitud de Palabra Conjunto de Prueba</b>
28	5.65% ± 1.88%	38.60% ± 5.11%	95.21% ± 1.90%	62.71% ± 5.09%
29	6.80% ± 2.76%	38.90% ± 3.46%	94.49% ± 2.25%	62.96% ± 2.87%
30	5.98% ± 2.19%	37.80% ± 5.69%	94.83% ± 2.21%	63.50% ± 6.06%
31	6.00% ± 2.15%	36.70% ± 3.51%	94.81% ± 1.95%	65.08% ± 3.84%
32	5.48% ± 1.41%	38.30% ± 5.67%	95.20% ± 1.40%	64.38% ± 5.32%
33	6.40% ± 3.26%	37.80% ± 5.06%	94.52% ± 2.90%	64.00% ± 5.07%
34	5.28% ± 2.03%	37.20% ± 4.83%	95.53% ± 1.90%	65.67% ± 3.77%
35	7.00% ± 3.71%	38.70% ± 5.78%	93.91% ± 3.63%	63.50% ± 5.20%
36	10.75% ± 18.89%	41.50% ± 12.99%	90.89% ± 15.71%	61.21% ± 10.62%
37	6.55% ± 2.82%	40.30% ± 4.55%	94.19% ± 2.55%	62.37% ± 4.16%
38	6.58% ± 2.49%	37.20% ± 6.37%	94.43% ± 2.21%	64.38% ± 6.02%
39	6.35% ± 3.64%	36.40% ± 4.08%	94.84% ± 2.83%	66.08% ± 3.10%
40	5.20% ± 2.22%	37.70% ± 4.41%	95.75% ± 1.94%	65.00% ± 3.87%
41	10.20% ± 19.01%	39.90% ± 13.00%	91.07% ± 15.87%	63.29% ± 11.12%
42	8.15% ± 3.91%	40.20% ± 5.98%	93.18% ± 3.21%	62.46% ± 5.88%
43	5.73% ± 3.72%	35.60% ± 5.26%	95.16% ± 3.52%	65.54% ± 5.82%
44	7.78% ± 4.19%	38.90% ± 7.99%	93.27% ± 3.72%	64.00% ± 7.62%
45	7.93% ± 4.55%	41.40% ± 4.64%	93.31% ± 3.99%	61.58% ± 4.11%
46	8.00% ± 4.45%	38.70% ± 5.92%	93.39% ± 3.60%	63.75% ± 5.16%
47	9.48% ± 19.13%	39.40% ± 12.65%	91.93% ± 15.94%	63.50% ± 9.67%
48	5.18% ± 2.25%	36.70% ± 5.52%	95.51% ± 2.07%	65.42% ± 5.08%
49	6.00% ± 5.94%	39.00% ± 6.94%	94.68% ± 5.77%	63.83% ± 7.03%
50	6.75% ± 2.98%	38.90% ± 3.75%	94.20% ± 2.65%	63.50% ± 2.82%
51	5.20% ± 3.63%	37.30% ± 4.51%	95.67% ± 3.48%	64.96% ± 3.70%
52	10.60% ± 18.98%	41.10% ± 12.52%	90.24% ± 15.93%	62.04% ± 9.61%
53	6.55% ± 4.73%	38.90% ± 5.67%	94.17% ± 4.53%	63.75% ± 4.76%
54	5.53% ± 3.31%	36.40% ± 5.72%	95.27% ± 3.09%	66.04% ± 4.90%
55	5.88% ± 4.22%	39.30% ± 4.51%	94.89% ± 4.00%	63.08% ± 3.84%
56	5.25% ± 2.22%	39.10% ± 4.13%	95.61% ± 2.08%	63.38% ± 4.14%
57	6.00% ± 3.34%	38.30% ± 4.91%	94.95% ± 3.04%	63.83% ± 5.10%
58	15.08% ± 25.81%	43.40% ± 16.43%	87.03% ± 21.49%	59.96% ± 12.75%
59	6.75% ± 3.29%	39.50% ± 4.76%	94.52% ± 2.69%	63.71% ± 3.78%
60	9.83% ± 7.06%	43.40% ± 7.11%	90.95% ± 7.31%	59.50% ± 6.82%
61	18.25% ± 31.32%	47.40% ± 18.91%	84.53% ± 26.15%	57.67% ± 14.71%
62	13.48% ± 19.19%	43.00% ± 12.97%	88.11% ± 16.21%	60.71% ± 10.21%
63	10.48% ± 19.36%	42.20% ± 12.60%	91.00% ± 16.30%	61.04% ± 9.96%
64	11.48% ± 19.84%	43.10% ± 12.41%	89.54% ± 17.15%	61.25% ± 10.31%
65	5.03% ± 3.74%	40.00% ± 4.98%	95.77% ± 3.44%	62.92% ± 4.92%
66	11.45% ± 18.84%	41.30% ± 12.64%	90.15% ± 15.68%	61.71% ± 9.95%
67	14.48% ± 26.11%	43.40% ± 16.78%	87.52% ± 21.71%	60.33% ± 12.88%
68	12.78% ± 18.78%	40.60% ± 13.09%	88.79% ± 15.61%	62.33% ± 10.44%
69	10.20% ± 19.28%	43.60% ± 12.32%	91.28% ± 16.09%	60.17% ± 9.43%
70	11.58% ± 18.97%	41.90% ± 12.25%	89.97% ± 15.86%	61.17% ± 9.82%
71	12.48% ± 19.30%	41.80% ± 12.21%	88.81% ± 16.55%	61.29% ± 9.16%
72	12.78% ± 18.87%	40.80% ± 12.67%	88.68% ± 15.88%	62.50% ± 10.05%

<b>Nodos</b>	<b>Error Conjunto entrenamiento</b>		<b>Error Conjunto de Prueba</b>		<b>Exactitud de Palabra Conjunto de entrenamiento</b>		<b>Exactitud de Palabra Conjunto de Prueba</b>	
73	6.25%	± 4.52%	40.30%	± 5.59%	94.71%	± 3.60%	62.83%	± 4.89%
74	6.53%	± 3.78%	39.00%	± 5.60%	94.39%	± 3.59%	63.54%	± 5.21%
75	12.33%	± 18.74%	42.70%	± 12.37%	89.07%	± 15.69%	60.83%	± 9.54%
76	6.15%	± 4.10%	37.50%	± 3.72%	94.68%	± 3.64%	65.42%	± 4.66%
77	10.73%	± 19.03%	42.20%	± 12.89%	91.14%	± 15.90%	61.54%	± 10.56%
78	13.35%	± 18.91%	42.20%	± 13.25%	88.12%	± 15.77%	60.67%	± 10.45%
79	8.55%	± 3.89%	40.40%	± 5.30%	92.50%	± 3.75%	63.63%	± 4.45%
80	11.80%	± 18.99%	42.10%	± 12.10%	90.23%	± 15.77%	61.04%	± 9.90%
81	7.75%	± 4.52%	42.00%	± 5.73%	93.30%	± 4.29%	61.13%	± 5.74%
82	12.15%	± 18.65%	41.50%	± 12.48%	89.87%	± 15.53%	61.75%	± 9.82%
83	6.23%	± 4.45%	39.90%	± 7.24%	94.56%	± 4.27%	63.46%	± 6.93%
84	20.38%	± 30.42%	47.60%	± 18.94%	82.57%	± 25.24%	57.25%	± 14.90%
85	7.35%	± 6.93%	41.70%	± 7.93%	93.45%	± 6.52%	61.00%	± 8.00%
86	11.88%	± 18.94%	42.00%	± 12.81%	90.17%	± 15.79%	61.62%	± 10.65%
87	13.43%	± 18.83%	42.80%	± 12.32%	88.15%	± 15.71%	60.87%	± 10.28%
88	14.15%	± 19.06%	44.30%	± 13.21%	87.90%	± 15.97%	59.29%	± 10.42%
89	7.90%	± 5.51%	41.00%	± 6.57%	93.06%	± 5.04%	62.04%	± 6.01%
90	6.88%	± 4.29%	40.60%	± 4.59%	94.05%	± 3.87%	63.29%	± 4.82%
91	12.25%	± 18.78%	42.30%	± 12.38%	88.24%	± 21.16%	59.75%	± 14.89%
92	17.43%	± 25.34%	46.10%	± 15.61%	85.49%	± 20.53%	57.42%	± 11.20%
93	12.48%	± 19.21%	43.80%	± 13.75%	88.77%	± 16.27%	59.67%	± 10.94%
94	8.88%	± 4.97%	41.20%	± 6.85%	92.33%	± 4.41%	62.17%	± 5.12%
95	16.53%	± 25.74%	47.20%	± 15.89%	85.66%	± 21.53%	57.67%	± 12.42%
96	10.03%	± 6.57%	42.70%	± 6.30%	90.71%	± 6.48%	61.38%	± 5.07%
97	19.20%	± 30.74%	46.20%	± 19.23%	82.63%	± 28.90%	57.38%	± 18.67%
98	12.70%	± 18.84%	43.30%	± 12.72%	88.24%	± 19.44%	59.21%	± 13.33%
99	9.55%	± 4.92%	41.00%	± 5.13%	91.60%	± 4.54%	61.67%	± 4.55%
100	20.83%	± 30.17%	48.00%	± 18.74%	82.01%	± 25.60%	55.38%	± 14.50%
101	7.05%	± 5.53%	40.30%	± 5.59%	93.67%	± 5.18%	62.75%	± 5.26%
102	18.53%	± 24.96%	48.40%	± 15.72%	83.80%	± 20.82%	56.33%	± 12.61%
103	20.50%	± 30.40%	47.20%	± 19.00%	81.08%	± 27.97%	56.63%	± 17.65%
104	14.68%	± 19.16%	44.80%	± 13.03%	86.62%	± 16.30%	58.13%	± 10.29%
105	10.10%	± 6.01%	42.80%	± 8.11%	90.99%	± 5.74%	60.75%	± 8.04%
106	9.73%	± 7.35%	40.80%	± 8.11%	91.58%	± 6.93%	62.12%	± 7.21%
107	32.90%	± 38.59%	54.70%	± 24.56%	70.26%	± 35.99%	49.75%	± 22.87%
108	10.78%	± 19.26%	41.40%	± 12.47%	89.99%	± 19.75%	61.25%	± 13.33%
109	28.13%	± 36.90%	51.40%	± 23.48%	76.29%	± 30.63%	53.79%	± 17.85%
110	9.90%	± 8.09%	41.40%	± 5.55%	90.98%	± 7.83%	61.29%	± 4.95%
111	13.25%	± 18.68%	43.80%	± 11.93%	88.23%	± 14.72%	60.13%	± 9.12%
112	24.45%	± 33.88%	50.20%	± 20.73%	78.57%	± 30.19%	54.75%	± 18.30%
113	21.68%	± 30.05%	48.80%	± 18.52%	81.47%	± 24.55%	56.75%	± 14.38%
114	29.20%	± 31.99%	53.50%	± 19.73%	74.00%	± 29.33%	50.29%	± 17.47%
115	19.05%	± 25.35%	46.80%	± 15.73%	81.93%	± 24.45%	55.63%	± 15.56%
116	16.60%	± 26.11%	45.40%	± 16.08%	84.93%	± 22.94%	58.88%	± 13.91%
117	8.25%	± 4.96%	41.10%	± 4.23%	92.77%	± 4.51%	61.08%	± 4.38%

Nodos	Error Conjunto entrenamiento	Error Conjunto de Prueba	Exactitud de Palabra Conjunto de entrenamiento	Exactitud de Palabra Conjunto de Prueba
118	14.38% ± 18.56%	43.90% ± 12.15%	87.67% ± 15.68%	60.42% ± 9.64%
119	20.05% ± 24.93%	47.00% ± 16.49%	79.66% ± 29.51%	54.67% ± 21.48%
120	13.33% ± 19.60%	43.60% ± 13.21%	87.25% ± 20.77%	60.29% ± 14.78%
121	26.60% ± 33.00%	52.70% ± 19.84%	74.16% ± 33.25%	49.96% ± 21.53%
122	18.30% ± 25.65%	47.80% ± 15.40%	83.27% ± 22.71%	55.63% ± 13.32%
123	16.33% ± 25.69%	46.80% ± 16.45%	84.32% ± 27.77%	55.67% ± 19.45%
124	31.28% ± 35.37%	53.00% ± 22.45%	73.44% ± 29.42%	53.50% ± 17.46%
125	14.98% ± 18.87%	45.40% ± 12.26%	86.57% ± 15.05%	57.92% ± 9.05%
126	19.40% ± 25.34%	46.20% ± 16.26%	83.08% ± 19.95%	57.50% ± 11.46%
127	17.00% ± 25.65%	45.30% ± 16.61%	84.29% ± 24.03%	58.08% ± 15.48%
128	27.13% ± 32.55%	51.00% ± 20.84%	76.67% ± 26.44%	55.08% ± 16.15%
129	20.03% ± 25.27%	46.60% ± 16.83%	82.59% ± 21.16%	57.13% ± 13.48%
130	26.88% ± 32.70%	51.60% ± 20.34%	74.18% ± 33.82%	50.33% ± 22.36%
131	27.05% ± 33.05%	49.20% ± 22.10%	74.44% ± 31.98%	54.42% ± 22.44%
132	20.63% ± 30.35%	48.80% ± 18.93%	81.44% ± 28.57%	54.67% ± 18.22%
133	23.03% ± 29.90%	50.80% ± 17.62%	80.33% ± 24.50%	55.08% ± 13.81%
134	18.43% ± 25.10%	45.80% ± 15.89%	81.49% ± 27.69%	56.25% ± 19.25%
135	28.48% ± 36.72%	51.90% ± 23.19%	72.13% ± 38.39%	49.83% ± 25.98%
136	20.75% ± 25.20%	48.50% ± 16.23%	81.65% ± 22.30%	56.13% ± 14.14%
137	31.93% ± 35.13%	55.30% ± 22.01%	68.40% ± 37.37%	46.96% ± 25.62%
138	26.38% ± 32.94%	51.70% ± 20.37%	75.07% ± 34.15%	51.33% ± 22.72%
139	26.18% ± 33.19%	49.70% ± 21.59%	72.58% ± 37.33%	50.00% ± 25.83%
140	37.38% ± 39.92%	57.70% ± 24.86%	66.52% ± 36.84%	47.79% ± 23.48%
141	28.75% ± 31.82%	53.50% ± 19.12%	75.14% ± 27.51%	51.58% ± 15.80%
142	17.05% ± 18.72%	45.40% ± 13.35%	85.13% ± 15.02%	58.79% ± 10.77%
143	19.10% ± 24.89%	48.60% ± 15.01%	83.27% ± 20.04%	55.88% ± 11.59%
144	24.88% ± 28.59%	51.80% ± 17.73%	78.41% ± 24.82%	52.50% ± 14.53%
145	19.80% ± 24.85%	48.40% ± 15.92%	80.70% ± 28.00%	53.42% ± 19.32%
146	31.70% ± 35.59%	54.30% ± 21.74%	72.18% ± 31.31%	50.63% ± 18.60%
147	23.30% ± 29.08%	49.80% ± 18.46%	77.69% ± 29.24%	53.13% ± 19.31%
148	28.85% ± 32.10%	51.30% ± 20.88%	74.64% ± 27.43%	53.92% ± 17.42%
149	35.25% ± 37.25%	57.10% ± 22.85%	67.55% ± 34.67%	47.38% ± 21.67%
150	37.50% ± 35.99%	57.30% ± 22.46%	63.25% ± 36.55%	44.46% ± 24.33%

Tabla E.0.3 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con k=5 y segmentación manual.

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
1	69.00% ± 2.72%	71.80% ± 2.50%	36.42% ± 1.34%	32.21% ± 3.70%
2	53.53% ± 6.80%	56.80% ± 6.85%	50.67% ± 7.55%	44.92% ± 7.28%
3	37.30% ± 11.64%	43.80% ± 9.77%	66.71% ± 10.05%	58.58% ± 8.67%
4	19.25% ± 6.09%	31.90% ± 5.13%	83.27% ± 5.30%	69.42% ± 5.24%
5	13.90% ± 2.70%	27.10% ± 4.92%	87.87% ± 3.09%	74.29% ± 4.62%
6	11.20% ± 2.57%	25.20% ± 4.51%	90.42% ± 2.38%	76.42% ± 4.05%

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
7	9.10% ± 1.39%	24.80% ± 3.97%	92.20% ± 1.19%	76.50% ± 2.97%
<b>10</b>	<b>5.75% ± 1.34%</b>	<b>24.30% ± 5.12%</b>	<b>95.42% ± 1.21%</b>	<b>78.25% ± 5.00%</b>
11	5.93% ± 1.40%	25.80% ± 4.44%	95.18% ± 1.43%	77.21% ± 3.99%
12	5.60% ± 1.05%	24.50% ± 4.58%	95.57% ± 1.02%	78.04% ± 3.67%
13	5.08% ± 1.00%	25.80% ± 4.40%	96.33% ± 0.82%	76.71% ± 4.31%
14	5.18% ± 0.95%	26.40% ± 4.19%	96.06% ± 0.91%	77.92% ± 3.25%
15	4.55% ± 1.13%	26.30% ± 5.00%	96.67% ± 1.18%	77.75% ± 3.77%
16	4.30% ± 0.66%	29.10% ± 4.18%	96.97% ± 0.72%	75.83% ± 3.59%
17	4.08% ± 0.88%	27.20% ± 3.00%	97.18% ± 0.80%	77.00% ± 3.58%
18	4.40% ± 1.12%	26.60% ± 4.90%	96.87% ± 0.89%	77.83% ± 4.14%
19	3.75% ± 0.68%	28.60% ± 3.90%	97.35% ± 0.71%	75.75% ± 3.07%
20	3.78% ± 0.73%	27.80% ± 3.49%	97.39% ± 0.61%	77.46% ± 2.99%
21	3.63% ± 1.06%	53.07% ± 7.00%	97.51% ± 0.99%	77.94% ± 2.85%
22	3.70% ± 0.86%	52.00% ± 8.82%	97.47% ± 0.73%	77.56% ± 3.98%
23	3.53% ± 0.48%	57.33% ± 6.87%	97.57% ± 0.43%	75.72% ± 2.88%
24	3.97% ± 1.37%	55.73% ± 7.92%	97.22% ± 1.18%	76.17% ± 3.71%
25	3.13% ± 0.90%	57.60% ± 9.30%	97.82% ± 0.84%	75.67% ± 5.06%
26	3.37% ± 0.48%	58.67% ± 7.51%	97.78% ± 0.43%	74.61% ± 3.51%
27	3.23% ± 0.56%	57.60% ± 7.83%	97.90% ± 0.52%	75.72% ± 3.56%
28	3.97% ± 0.88%	59.20% ± 9.22%	97.28% ± 0.84%	74.89% ± 4.17%
29	3.13% ± 0.69%	56.80% ± 9.10%	97.87% ± 0.54%	76.06% ± 5.05%
30	3.40% ± 0.83%	57.33% ± 10.55%	97.78% ± 0.68%	75.72% ± 5.36%
31	3.37% ± 0.64%	62.93% ± 10.42%	97.75% ± 0.60%	72.33% ± 5.69%
32	3.40% ± 0.81%	59.20% ± 8.31%	97.68% ± 0.65%	75.06% ± 4.67%
33	3.23% ± 0.82%	55.47% ± 4.75%	97.88% ± 0.76%	78.00% ± 2.83%
34	2.90% ± 0.89%	58.67% ± 7.66%	98.13% ± 0.85%	75.72% ± 4.34%
35	2.93% ± 0.56%	55.73% ± 7.63%	98.18% ± 0.53%	76.44% ± 4.30%
36	3.83% ± 2.27%	59.47% ± 6.74%	97.40% ± 1.97%	75.00% ± 3.07%
37	3.47% ± 0.83%	58.13% ± 8.26%	97.64% ± 0.87%	75.39% ± 4.44%
38	2.97% ± 0.79%	54.93% ± 6.50%	98.22% ± 0.68%	76.72% ± 3.14%
39	3.33% ± 0.84%	58.13% ± 8.80%	97.71% ± 0.77%	75.72% ± 4.25%
40	2.80% ± 0.80%	56.27% ± 7.32%	98.24% ± 0.66%	77.11% ± 3.40%
41	3.17% ± 0.65%	57.07% ± 8.21%	97.97% ± 0.63%	76.00% ± 3.72%
42	3.87% ± 2.67%	57.07% ± 10.85%	97.18% ± 3.06%	75.33% ± 5.81%
43	3.07% ± 0.68%	56.27% ± 5.34%	98.10% ± 0.62%	76.78% ± 3.24%
44	3.50% ± 2.71%	58.67% ± 7.81%	97.83% ± 2.05%	75.56% ± 4.44%
45	2.87% ± 0.72%	54.40% ± 9.77%	98.24% ± 0.62%	76.72% ± 4.55%
46	3.47% ± 2.36%	57.33% ± 8.09%	97.64% ± 2.01%	76.39% ± 4.31%
47	2.83% ± 0.65%	53.87% ± 8.80%	98.28% ± 0.46%	77.61% ± 4.42%

<b>Nodos</b>	<b>Error Conjunto de entrenamiento</b>		<b>Error Conjunto de prueba</b>		<b>Exactitud de palabra Conjunto entrenamiento</b>		<b>Exactitud de palabra Conjunto de prueba</b>	
48	3.63%	± 2.68%	62.13%	± 7.54%	97.44%	± 2.69%	75.33%	± 4.62%
49	3.43%	± 2.43%	57.60%	± 6.38%	97.79%	± 1.77%	75.78%	± 3.41%
50	2.80%	± 0.59%	59.47%	± 8.53%	98.26%	± 0.53%	75.00%	± 3.87%
51	2.90%	± 0.93%	59.47%	± 5.21%	98.22%	± 0.95%	75.39%	± 1.86%
52	3.67%	± 3.35%	57.60%	± 7.53%	97.22%	± 4.05%	75.39%	± 4.07%
53	2.87%	± 1.03%	56.27%	± 6.32%	98.17%	± 0.93%	77.06%	± 2.73%
54	3.33%	± 1.38%	60.80%	± 8.31%	97.87%	± 1.14%	74.56%	± 4.61%
55	3.30%	± 1.01%	58.67%	± 7.04%	97.92%	± 1.00%	75.61%	± 3.68%
56	2.87%	± 0.48%	54.67%	± 6.87%	98.15%	± 0.48%	77.89%	± 3.66%
57	3.37%	± 2.44%	57.87%	± 8.26%	97.86%	± 1.88%	77.00%	± 4.06%
58	3.43%	± 0.94%	56.00%	± 8.14%	97.62%	± 0.87%	77.44%	± 3.68%
59	3.07%	± 0.70%	57.60%	± 6.20%	98.08%	± 0.65%	76.83%	± 3.92%
60	3.63%	± 1.96%	60.27%	± 7.32%	97.68%	± 1.51%	74.83%	± 3.58%
61	3.00%	± 0.91%	58.93%	± 7.32%	98.10%	± 0.73%	75.33%	± 3.66%
62	3.17%	± 0.70%	59.47%	± 7.39%	97.94%	± 0.52%	74.61%	± 3.98%
63	3.00%	± 1.00%	59.20%	± 6.45%	98.15%	± 0.92%	75.11%	± 3.91%
64	2.80%	± 0.92%	56.80%	± 9.82%	98.26%	± 0.79%	76.33%	± 4.65%
65	2.77%	± 0.62%	55.20%	± 6.79%	98.38%	± 0.48%	77.44%	± 4.11%
66	2.73%	± 0.46%	55.20%	± 10.28%	98.29%	± 0.32%	78.28%	± 5.06%
67	3.10%	± 2.61%	57.60%	± 8.11%	98.01%	± 2.06%	76.83%	± 4.13%
68	3.13%	± 0.90%	57.60%	± 7.38%	97.99%	± 0.73%	76.61%	± 4.28%
69	3.27%	± 2.46%	59.47%	± 6.74%	98.00%	± 1.86%	75.00%	± 3.49%
70	3.47%	± 1.14%	57.87%	± 5.21%	97.71%	± 1.17%	76.89%	± 3.41%
71	3.53%	± 2.59%	59.73%	± 6.13%	97.54%	± 2.37%	76.72%	± 3.40%
72	3.93%	± 2.96%	60.27%	± 7.32%	97.07%	± 3.47%	74.11%	± 5.48%
73	3.30%	± 2.60%	58.40%	± 10.23%	97.64%	± 3.19%	76.22%	± 5.82%
74	3.73%	± 2.48%	60.00%	± 8.00%	97.46%	± 2.39%	75.39%	± 3.00%
75	3.47%	± 2.49%	61.07%	± 7.92%	97.60%	± 2.75%	75.67%	± 4.20%
76	2.90%	± 0.60%	52.27%	± 5.34%	98.17%	± 0.46%	79.67%	± 3.05%
77	3.87%	± 2.36%	59.20%	± 10.05%	97.42%	± 1.93%	76.06%	± 4.86%
78	3.13%	± 0.77%	60.27%	± 5.95%	98.07%	± 0.64%	75.28%	± 2.63%
79	3.07%	± 1.27%	57.07%	± 4.13%	98.11%	± 1.10%	77.28%	± 2.41%
80	4.03%	± 2.29%	57.60%	± 6.38%	97.32%	± 1.84%	76.39%	± 3.60%
81	4.13%	± 3.48%	60.00%	± 7.25%	97.11%	± 3.24%	76.33%	± 3.30%
82	3.83%	± 2.66%	58.13%	± 4.75%	97.63%	± 2.04%	77.00%	± 1.96%
83	4.03%	± 3.34%	62.67%	± 10.55%	96.90%	± 3.82%	74.94%	± 4.96%
84	3.43%	± 2.68%	60.00%	± 8.42%	97.68%	± 2.68%	75.72%	± 4.24%
85	3.33%	± 1.08%	61.07%	± 6.50%	97.82%	± 1.02%	75.22%	± 3.31%
86	2.83%	± 0.88%	56.27%	± 6.67%	98.33%	± 0.80%	78.11%	± 2.96%



Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
87	3.07% ± 0.92%	56.80% ± 5.89%	98.10% ± 0.84%	77.44% ± 2.81%
88	6.27% ± 4.54%	62.93% ± 6.13%	95.28% ± 4.33%	74.50% ± 4.08%
89	3.57% ± 2.43%	54.93% ± 9.38%	97.78% ± 1.84%	78.56% ± 4.27%
90	3.87% ± 2.72%	58.93% ± 7.00%	97.19% ± 2.99%	75.56% ± 3.94%
91	4.33% ± 3.85%	62.93% ± 9.25%	96.46% ± 4.33%	74.00% ± 5.63%
92	3.80% ± 2.62%	60.00% ± 7.09%	97.36% ± 2.70%	76.17% ± 3.97%
93	3.20% ± 0.68%	60.00% ± 7.09%	98.01% ± 0.60%	75.11% ± 3.53%
94	4.73% ± 3.36%	60.80% ± 7.44%	96.11% ± 4.28%	74.72% ± 4.19%
95	4.07% ± 3.59%	56.53% ± 8.26%	96.90% ± 4.01%	77.39% ± 4.82%
96	4.03% ± 2.36%	60.53% ± 7.39%	97.26% ± 1.79%	76.22% ± 2.88%
97	6.00% ± 4.46%	63.47% ± 8.93%	95.11% ± 4.64%	74.06% ± 4.34%
98	4.30% ± 3.28%	61.33% ± 7.51%	96.72% ± 3.59%	74.83% ± 4.72%
99	4.77% ± 3.96%	60.80% ± 8.31%	96.17% ± 4.32%	75.61% ± 3.86%
100	4.53% ± 5.29%	60.80% ± 12.39%	96.79% ± 4.56%	74.72% ± 5.00%

**Tabla E.0.4 Porcentaje de error promedio y exactitud de palabra con sus respectivas desviaciones estándar obtenidos variando el número de nodos en la capa oculta de la red FF para el reconocimiento de palabras. Segmentación automática.**

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
5	87.30% ± 5.59%	88.80% ± 2.28%	12.28% ± 10.93%	11.00% ± 9.36%
10	86.28% ± 6.53%	88.80% ± 2.28%	17.01% ± 12.06%	14.75% ± 10.92%
15	83.93% ± 9.35%	87.10% ± 5.41%	17.38% ± 11.13%	14.46% ± 9.52%
20	84.85% ± 6.26%	87.80% ± 3.30%	19.99% ± 10.21%	17.04% ± 8.81%
25	85.35% ± 8.81%	88.00% ± 3.78%	16.07% ± 14.34%	13.42% ± 11.98%
<b>30</b>	<b>82.30% ± 9.94%</b>	<b>86.10% ± 5.78%</b>	<b>19.92% ± 13.22%</b>	<b>16.75% ± 10.92%</b>
35	88.15% ± 5.66%	89.00% ± 3.34%	18.53% ± 11.31%	17.63% ± 10.01%
40	87.80% ± 8.53%	89.00% ± 4.03%	14.59% ± 13.44%	13.54% ± 10.59%
45	86.85% ± 7.85%	88.60% ± 3.68%	14.04% ± 11.92%	12.13% ± 9.00%
50	88.45% ± 5.09%	89.50% ± 1.57%	11.61% ± 10.83%	10.46% ± 9.54%
55	90.00% ± 0.00%	90.00% ± 0.00%	12.29% ± 10.60%	12.29% ± 10.60%
60	90.00% ± 0.00%	90.00% ± 0.00%	10.63% ± 11.43%	10.63% ± 11.43%
65	88.68% ± 5.93%	89.60% ± 1.79%	14.04% ± 12.29%	13.00% ± 10.84%
70	90.00% ± 0.00%	90.00% ± 0.00%	13.75% ± 10.65%	13.75% ± 10.65%
75	90.00% ± 0.00%	90.00% ± 0.00%	11.04% ± 10.50%	11.04% ± 10.50%
80	90.00% ± 0.00%	90.00% ± 0.00%	11.67% ± 12.51%	11.67% ± 12.51%
85	90.00% ± 0.00%	90.00% ± 0.00%	12.50% ± 9.27%	12.50% ± 9.27%
90	90.00% ± 0.00%	90.00% ± 0.00%	9.79% ± 8.14%	9.79% ± 8.14%
95	90.00% ± 0.00%	90.00% ± 0.00%	11.88% ± 8.47%	11.88% ± 8.47%
100	90.00% ± 0.00%	90.00% ± 0.00%	11.67% ± 9.80%	11.67% ± 9.80%
105	90.00% ± 0.00%	90.00% ± 0.00%	11.04% ± 9.87%	11.04% ± 9.87%
110	90.00% ± 0.00%	90.00% ± 0.00%	11.46% ± 10.01%	11.46% ± 10.01%

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
115	90.00% ± 0.00%	90.00% ± 0.00%	7.92% ± 8.21%	7.92% ± 8.21%
120	90.00% ± 0.00%	90.00% ± 0.00%	12.50% ± 9.46%	12.50% ± 9.46%
125	90.00% ± 0.00%	90.00% ± 0.00%	14.38% ± 9.60%	14.38% ± 9.60%
130	90.00% ± 0.00%	90.00% ± 0.00%	11.04% ± 10.84%	11.04% ± 10.84%
135	90.00% ± 0.00%	90.00% ± 0.00%	11.88% ± 9.87%	11.88% ± 9.87%
140	90.00% ± 0.00%	90.00% ± 0.00%	13.13% ± 10.05%	13.13% ± 10.05%
145	90.00% ± 0.00%	90.00% ± 0.00%	11.88% ± 9.10%	11.88% ± 9.10%
150	90.00% ± 0.00%	90.00% ± 0.00%	14.58% ± 10.34%	14.58% ± 10.34%

Tabla E.0.5 Exactitud de palabra y porcentaje de error de reconocimiento promedio la red SOM-FF con k=5 y segmentación automática.

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
5	31.40% ± 4.67%	58.30% ± 5.63%	71.17% ± 4.26%	43.08% ± 6.73%
10	19.98% ± 1.59%	53.40% ± 4.99%	81.60% ± 1.53%	48.75% ± 5.14%
15	15.20% ± 2.24%	49.20% ± 5.00%	86.15% ± 2.39%	53.13% ± 5.64%
20	13.95% ± 2.93%	50.10% ± 5.25%	87.01% ± 3.00%	50.79% ± 6.10%
25	12.10% ± 2.42%	50.20% ± 6.22%	88.76% ± 2.60%	51.96% ± 5.40%
30	11.60% ± 2.56%	50.40% ± 5.01%	89.16% ± 2.44%	51.83% ± 6.25%
35	11.50% ± 3.62%	49.10% ± 5.25%	89.29% ± 3.26%	53.00% ± 5.43%
<b>40</b>	<b>12.40% ± 3.66%</b>	<b>48.70% ± 5.08%</b>	<b>88.34% ± 3.58%</b>	<b>53.88% ± 5.05%</b>
45	11.75% ± 3.91%	49.70% ± 5.36%	89.01% ± 3.95%	52.67% ± 4.92%
50	11.55% ± 4.05%	51.20% ± 6.66%	89.28% ± 4.31%	50.29% ± 5.99%
55	10.10% ± 3.30%	49.90% ± 6.57%	90.83% ± 2.67%	51.87% ± 7.28%
60	19.58% ± 24.54%	52.00% ± 13.76%	81.24% ± 24.01%	50.00% ± 13.78%
65	25.98% ± 32.97%	57.10% ± 17.36%	74.09% ± 35.07%	43.79% ± 20.02%
70	31.10% ± 35.12%	59.80% ± 18.29%	70.11% ± 35.41%	42.88% ± 19.99%
75	32.75% ± 34.19%	59.70% ± 18.71%	72.14% ± 27.25%	45.88% ± 12.57%
80	26.33% ± 32.97%	59.30% ± 16.29%	73.82% ± 34.38%	41.50% ± 18.24%
85	42.68% ± 39.79%	65.80% ± 21.29%	60.51% ± 37.61%	37.75% ± 20.29%
90	40.55% ± 38.08%	64.00% ± 20.62%	61.35% ± 36.74%	39.04% ± 20.51%
95	51.15% ± 39.95%	69.60% ± 21.27%	51.66% ± 38.19%	34.00% ± 20.71%
100	59.43% ± 38.70%	73.00% ± 21.65%	49.59% ± 32.26%	36.42% ± 15.93%
105	64.30% ± 36.24%	76.50% ± 19.53%	37.91% ± 36.40%	26.08% ± 21.02%
110	70.65% ± 34.45%	79.60% ± 18.57%	33.99% ± 32.95%	25.62% ± 19.07%
115	63.95% ± 36.64%	76.10% ± 19.81%	39.56% ± 35.44%	27.46% ± 19.26%
120	70.18% ± 35.33%	79.70% ± 18.63%	36.18% ± 32.85%	26.54% ± 17.16%
125	78.58% ± 27.97%	83.90% ± 15.20%	23.66% ± 28.89%	18.75% ± 18.35%
130	62.98% ± 38.01%	75.60% ± 20.56%	39.93% ± 37.75%	27.42% ± 21.20%

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
135	75.80% ± 29.37%	83.10% ± 14.28%	24.84% ± 30.93%	17.33% ± 16.55%
140	79.28% ± 26.36%	84.90% ± 12.47%	26.30% ± 26.36%	20.71% ± 14.15%
145	86.23% ± 16.76%	88.20% ± 8.05%	18.48% ± 18.47%	16.54% ± 11.97%
150	78.70% ± 27.87%	83.90% ± 15.13%	24.20% ± 28.64%	19.13% ± 17.40%

Tabla E.0.6 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con k=5 y segmentación automática.

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
10	28.75% ± 8.18%	52.80% ± 9.67%	74.56% ± 8.16%	49.58% ± 10.94%
15	15.70% ± 1.83%	46.60% ± 6.47%	86.42% ± 2.17%	56.83% ± 7.29%
20	8.00% ± 1.86%	44.20% ± 4.85%	93.12% ± 1.79%	60.58% ± 5.95%
25	6.00% ± 2.13%	49.20% ± 7.50%	94.73% ± 2.06%	54.25% ± 6.81%
30	3.05% ± 1.21%	46.40% ± 3.86%	97.54% ± 1.10%	58.92% ± 3.58%
35	2.55% ± 1.04%	47.20% ± 3.79%	97.81% ± 0.94%	58.83% ± 5.53%
40	2.30% ± 0.67%	43.80% ± 6.83%	97.92% ± 0.84%	60.75% ± 7.17%
45	2.61% ± 1.05%	47.56% ± 4.22%	97.94% ± 0.89%	57.78% ± 4.53%
50	1.61% ± 0.49%	46.00% ± 4.24%	98.68% ± 0.45%	59.81% ± 4.69%
55	1.94% ± 0.85%	44.00% ± 2.24%	98.40% ± 0.74%	61.02% ± 2.94%
60	1.61% ± 0.86%	42.89% ± 2.85%	98.84% ± 0.70%	61.30% ± 3.34%
65	2.00% ± 0.90%	44.67% ± 6.08%	98.45% ± 0.81%	62.13% ± 6.81%
70	1.83% ± 0.66%	45.78% ± 5.33%	98.59% ± 0.59%	59.17% ± 4.43%
75	1.89% ± 0.89%	44.89% ± 3.18%	98.45% ± 0.88%	61.20% ± 3.64%
80	1.63% ± 0.52%	44.50% ± 4.11%	98.78% ± 0.34%	60.31% ± 3.08%
85	2.00% ± 0.76%	46.75% ± 3.99%	98.49% ± 0.77%	60.63% ± 4.15%
90	5.50% ± 4.65%	42.00% ± 5.54%	95.54% ± 4.25%	64.52% ± 4.76%
95	2.30% ± 1.96%	45.20% ± 7.69%	98.13% ± 1.97%	59.17% ± 7.45%
<b>100</b>	<b>1.40% ± 0.42%</b>	<b>40.00% ± 4.69%</b>	<b>98.88% ± 0.52%</b>	<b>67.70% ± 2.58%</b>
105	1.70% ± 0.27%	39.20% ± 1.79%	98.71% ± 0.40%	67.67% ± 1.37%
110	1.75% ± 0.65%	46.00% ± 2.31%	98.65% ± 0.71%	61.04% ± 4.58%
115	3.75% ± 4.84%	39.50% ± 5.74%	96.93% ± 4.07%	65.62% ± 6.88%
120	1.75% ± 0.35%	49.00% ± 1.41%	98.44% ± 0.15%	54.58% ± 1.77%
125	2.60% ± 4.14%	42.00% ± 3.16%	98.04% ± 3.46%	62.33% ± 4.27%
130	3.88% ± 4.42%	47.00% ± 3.46%	97.60% ± 2.58%	58.33% ± 4.95%
135	2.88% ± 2.87%	44.50% ± 6.19%	97.50% ± 3.24%	59.17% ± 7.07%
140	4.50% ± 5.70%	44.00% ± 4.32%	96.25% ± 5.17%	62.29% ± 4.53%
145	4.38% ± 4.09%	43.50% ± 5.00%	96.35% ± 3.56%	61.46% ± 5.83%

**Tabla E.0.7 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red FF con segmentación automática y utilizando el número de elementos que componen a la secuencia como característica.**

<b>Nodos</b>	<b>Error Conjunto de entrenamiento</b>	<b>Error Conjunto de prueba</b>	<b>Exactitud de palabra Conjunto entrenamiento</b>	<b>Exactitud de palabra Conjunto de prueba</b>
5	84.33% ± 10.66%	87.90% ± 4.23%	17.50% ± 13.88%	13.67% ± 9.34%
10	87.98% ± 6.41%	89.20% ± 3.14%	14.57% ± 12.21%	13.46% ± 11.30%
15	86.00% ± 8.88%	88.60% ± 3.68%	15.58% ± 14.09%	13.38% ± 11.26%
<b>20</b>	<b>83.30% ± 9.08%</b>	<b>85.90% ± 5.89%</b>	<b>19.13% ± 10.89%</b>	<b>16.67% ± 9.58%</b>
25	82.90% ± 7.80%	86.70% ± 4.37%	15.91% ± 10.02%	11.92% ± 8.69%
30	86.45% ± 5.48%	88.40% ± 2.95%	17.57% ± 9.44%	15.83% ± 8.36%
35	87.58% ± 6.06%	89.10% ± 2.55%	16.14% ± 11.27%	14.87% ± 10.36%
40	86.73% ± 6.86%	88.50% ± 3.10%	17.31% ± 12.15%	15.46% ± 11.09%
45	90.00% ± 0.00%	90.00% ± 0.00%	12.92% ± 9.74%	12.92% ± 9.74%
50	87.80% ± 7.44%	88.60% ± 4.73%	15.54% ± 12.51%	14.71% ± 11.01%
55	89.63% ± 1.68%	89.70% ± 1.34%	13.23% ± 10.41%	13.17% ± 10.41%
60	90.00% ± 0.00%	90.00% ± 0.00%	14.17% ± 11.74%	14.17% ± 11.74%
65	90.00% ± 0.00%	90.00% ± 0.00%	12.92% ± 9.45%	12.92% ± 9.45%
70	90.00% ± 0.00%	90.00% ± 0.00%	13.13% ± 9.39%	13.13% ± 9.39%
75	90.00% ± 0.00%	90.00% ± 0.00%	12.08% ± 10.81%	12.08% ± 10.81%
80	90.00% ± 0.00%	90.00% ± 0.00%	13.13% ± 11.49%	13.13% ± 11.49%
85	90.00% ± 0.00%	90.00% ± 0.00%	11.46% ± 9.83%	11.46% ± 9.83%
90	90.00% ± 0.00%	90.00% ± 0.00%	12.29% ± 10.34%	12.29% ± 10.34%
95	90.00% ± 0.00%	90.00% ± 0.00%	12.50% ± 8.86%	12.50% ± 8.86%
100	90.00% ± 0.00%	90.00% ± 0.00%	12.08% ± 10.64%	12.08% ± 10.64%
105	90.00% ± 0.00%	90.00% ± 0.00%	17.08% ± 10.72%	17.08% ± 10.72%
110	90.00% ± 0.00%	90.00% ± 0.00%	7.08% ± 6.91%	7.08% ± 6.91%
115	90.00% ± 0.00%	90.00% ± 0.00%	11.46% ± 10.01%	11.46% ± 10.01%
120	90.00% ± 0.00%	90.00% ± 0.00%	15.42% ± 10.82%	15.42% ± 10.82%
125	90.00% ± 0.00%	90.00% ± 0.00%	13.75% ± 9.38%	13.75% ± 9.38%
130	90.00% ± 0.00%	90.00% ± 0.00%	13.13% ± 9.96%	13.13% ± 9.96%
135	90.00% ± 0.00%	90.00% ± 0.00%	10.63% ± 11.27%	10.63% ± 11.27%
140	90.00% ± 0.00%	90.00% ± 0.00%	11.88% ± 9.59%	11.88% ± 9.59%
145	90.00% ± 0.00%	90.00% ± 0.00%	14.37% ± 9.51%	14.37% ± 9.51%
150	90.00% ± 0.00%	90.00% ± 0.00%	11.46% ± 9.16%	11.46% ± 9.16%

**Tabla E.0.8 Porcentaje Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando la red SOM-FF con k=5, segmentación automática y utilizando el número de elementos que componen a la secuencia como característica.**

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
5	31.38% ± 3.91%	58.60% ± 4.68%	71.36% ± 3.93%	43.42% ± 5.31%
10	18.43% ± 2.13%	53.50% ± 7.89%	83.07% ± 1.84%	49.08% ± 7.61%
15	14.65% ± 2.11%	51.30% ± 5.28%	86.38% ± 1.99%	50.96% ± 5.63%
20	12.93% ± 1.77%	52.20% ± 6.61%	88.14% ± 1.71%	49.50% ± 6.35%
25	13.00% ± 2.62%	50.40% ± 6.11%	87.74% ± 2.40%	52.17% ± 5.69%
30	10.33% ± 2.78%	50.90% ± 6.60%	90.63% ± 2.56%	51.46% ± 7.55%
35	10.93% ± 2.91%	50.40% ± 5.79%	89.89% ± 2.82%	50.83% ± 5.47%
40	11.28% ± 3.26%	48.90% ± 5.64%	89.20% ± 3.27%	53.13% ± 6.25%
45	15.78% ± 17.68%	52.30% ± 10.67%	85.08% ± 18.23%	50.75% ± 11.59%
<b>50</b>	<b>12.15% ± 5.04%</b>	<b>48.50% ± 5.10%</b>	<b>89.05% ± 4.66%</b>	<b>54.96% ± 4.80%</b>
55	22.65% ± 29.34%	54.60% ± 15.78%	79.20% ± 27.13%	48.62% ± 14.73%
60	20.18% ± 24.25%	54.50% ± 12.93%	82.58% ± 19.31%	49.92% ± 8.70%
65	19.20% ± 24.45%	53.90% ± 13.15%	82.84% ± 20.06%	50.54% ± 9.99%
70	47.48% ± 39.52%	67.70% ± 20.91%	53.17% ± 39.77%	33.46% ± 21.83%
75	23.90% ± 28.77%	55.80% ± 15.68%	77.18% ± 27.73%	47.25% ± 15.47%
80	30.80% ± 35.33%	59.90% ± 18.62%	70.35% ± 34.96%	42.04% ± 19.07%
85	35.90% ± 36.56%	61.90% ± 19.12%	67.09% ± 33.72%	41.67% ± 17.40%
90	41.83% ± 40.58%	64.70% ± 21.62%	58.32% ± 41.58%	36.04% ± 23.61%
95	42.58% ± 39.96%	66.00% ± 20.49%	57.53% ± 40.96%	35.37% ± 23.11%
100	47.20% ± 39.92%	68.50% ± 20.45%	51.84% ± 42.43%	30.87% ± 23.50%
105	51.50% ± 39.82%	70.20% ± 20.73%	50.61% ± 39.46%	31.63% ± 20.79%
110	79.23% ± 26.62%	84.60% ± 13.25%	23.56% ± 26.82%	18.25% ± 14.77%
115	62.98% ± 37.88%	74.20% ± 22.43%	41.48% ± 35.94%	30.00% ± 20.66%
120	56.15% ± 38.61%	72.30% ± 20.25%	48.94% ± 35.67%	33.25% ± 18.88%
125	72.30% ± 31.52%	81.30% ± 15.62%	29.84% ± 31.52%	21.08% ± 17.00%
130	86.73% ± 14.65%	88.50% ± 6.71%	18.44% ± 17.12%	16.92% ± 12.31%
135	75.60% ± 29.61%	82.70% ± 15.00%	29.39% ± 29.31%	22.42% ± 16.37%
140	78.95% ± 27.09%	84.10% ± 14.65%	22.21% ± 28.65%	17.04% ± 17.48%
145	86.33% ± 16.44%	88.40% ± 7.16%	16.80% ± 19.07%	14.67% ± 12.74%
150	75.88% ± 29.24%	82.30% ± 16.24%	27.03% ± 29.75%	20.75% ± 18.13%

**Tabla E.0.9 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con k=5, segmentación automática y utilizando el número de elementos que componen a la secuencia como característica.**

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
10	34.00% ± 7.07%	46.83% ± 6.95%	66.88% ± 8.68%	54.58% ± 8.44%
15	19.42% ± 7.87%	41.00% ± 8.76%	81.32% ± 8.85%	60.62% ± 9.76%
20	9.29% ± 4.56%	39.33% ± 4.46%	91.35% ± 4.29%	63.06% ± 4.09%
25	5.42% ± 1.61%	35.67% ± 6.49%	95.19% ± 1.78%	67.92% ± 5.32%
30	3.63% ± 1.35%	36.67% ± 5.61%	96.98% ± 1.24%	66.81% ± 5.17%
35	3.04% ± 0.66%	37.17% ± 5.15%	97.34% ± 0.78%	66.04% ± 4.82%
40	2.88% ± 1.35%	35.00% ± 5.22%	97.47% ± 1.36%	69.10% ± 5.23%
45	1.95% ± 1.11%	34.36% ± 4.46%	98.33% ± 1.01%	69.02% ± 4.03%
50	2.95% ± 1.06%	33.82% ± 5.47%	97.50% ± 1.13%	68.33% ± 4.93%
55	1.55% ± 0.85%	33.45% ± 4.11%	98.73% ± 0.86%	69.62% ± 3.50%
60	1.50% ± 0.45%	32.91% ± 3.27%	98.67% ± 0.54%	69.39% ± 4.00%
65	1.50% ± 1.27%	35.00% ± 4.74%	98.75% ± 1.19%	68.33% ± 3.95%
70	0.85% ± 0.58%	30.00% ± 3.53%	99.33% ± 0.55%	73.25% ± 3.46%
75	1.40% ± 0.81%	31.60% ± 2.80%	98.73% ± 0.85%	72.08% ± 3.29%
<b>80</b>	<b>1.65% ± 0.58%</b>	<b>29.60% ± 4.09%</b>	<b>98.67% ± 0.57%</b>	<b>73.92% ± 3.09%</b>
85	1.50% ± 0.78%	29.60% ± 4.88%	98.81% ± 0.63%	73.75% ± 4.24%
90	0.89% ± 0.74%	38.89% ± 5.75%	99.35% ± 0.59%	66.48% ± 5.92%
95	1.07% ± 1.21%	38.00% ± 3.65%	99.32% ± 0.84%	67.50% ± 2.55%
100	2.00% ± 3.76%	38.29% ± 2.69%	98.30% ± 3.76%	66.90% ± 2.79%

**Tabla E.0.10 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con k=1, sin utilizar la función de probabilidad y con segmentación automática.**

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
10	35.80% ± 10.20%	51.60% ± 3.85%	67.75% ± 12.02%	52.17% ± 7.11%
15	13.90% ± 2.53%	49.20% ± 6.42%	88.00% ± 2.81%	55.83% ± 5.30%
20	9.80% ± 2.44%	51.20% ± 8.07%	92.17% ± 1.70%	54.67% ± 8.39%
25	5.70% ± 1.52%	49.60% ± 3.29%	95.21% ± 1.47%	56.33% ± 4.39%
30	5.10% ± 1.14%	44.00% ± 2.00%	96.00% ± 0.95%	60.00% ± 2.76%
35	2.90% ± 0.42%	47.20% ± 3.03%	97.83% ± 0.68%	57.00% ± 3.26%
40	2.30% ± 0.76%	53.60% ± 2.61%	98.08% ± 0.79%	53.67% ± 2.47%
45	1.90% ± 1.08%	48.40% ± 6.23%	98.46% ± 0.96%	56.67% ± 6.80%
50	2.10% ± 0.96%	47.20% ± 3.35%	98.17% ± 1.01%	58.67% ± 2.09%
55	2.00% ± 0.61%	46.00% ± 4.90%	98.50% ± 0.86%	59.00% ± 4.06%
60	1.30% ± 0.57%	49.60% ± 4.56%	99.08% ± 0.50%	56.50% ± 4.39%

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
65	2.20% ± 0.45%	45.60% ± 4.98%	98.08% ± 0.45%	58.17% ± 4.31%
70	2.20% ± 1.04%	44.00% ± 4.00%	98.17% ± 1.09%	61.00% ± 4.31%
75	2.40% ± 0.96%	45.20% ± 3.35%	98.21% ± 0.87%	60.50% ± 3.42%
<b>80</b>	<b>2.30% ± 1.10%</b>	<b>43.60% ± 3.85%</b>	<b>98.21% ± 1.18%</b>	<b>62.17% ± 4.31%</b>
85	1.60% ± 0.65%	47.20% ± 7.69%	98.58% ± 0.74%	58.17% ± 8.47%
90	2.00% ± 0.61%	46.00% ± 4.24%	98.33% ± 0.49%	58.83% ± 5.06%
95	2.20% ± 0.45%	44.40% ± 6.99%	98.17% ± 0.56%	62.00% ± 6.17%
100	1.60% ± 0.55%	45.20% ± 3.90%	98.96% ± 0.57%	61.33% ± 3.47%

Tabla E.0.11 Exactitud de palabra y porcentaje de error de reconocimiento promedio utilizando el método propuesto con k=5, sin utilizar la función de probabilidad y con segmentación automática.

Nodos	Error Conjunto de entrenamiento	Error Conjunto de prueba	Exactitud de palabra Conjunto entrenamiento	Exactitud de palabra Conjunto de prueba
10	32.80% ± 12.80%	56.00% ± 9.38%	72.63% ± 10.20%	49.00% ± 6.16%
15	12.80% ± 1.20%	47.60% ± 3.85%	88.71% ± 1.64%	55.50% ± 3.80%
20	7.70% ± 2.08%	52.40% ± 3.29%	93.13% ± 2.24%	50.00% ± 1.67%
25	5.70% ± 2.39%	47.20% ± 9.76%	94.96% ± 2.17%	56.83% ± 10.53%
30	2.60% ± 1.14%	46.80% ± 3.63%	97.50% ± 1.07%	58.33% ± 2.57%
35	2.40% ± 2.04%	43.60% ± 3.29%	97.88% ± 1.87%	60.50% ± 3.56%
40	1.80% ± 0.76%	52.00% ± 2.83%	98.50% ± 0.74%	51.83% ± 3.70%
45	3.50% ± 1.41%	47.20% ± 4.38%	96.88% ± 1.18%	58.00% ± 6.47%
50	1.60% ± 0.65%	44.00% ± 7.07%	98.54% ± 0.66%	60.17% ± 6.81%
55	1.30% ± 0.45%	44.00% ± 4.24%	98.92% ± 0.48%	62.17% ± 3.94%
60	1.30% ± 0.76%	46.40% ± 1.67%	99.13% ± 0.68%	59.00% ± 3.08%
65	1.20% ± 0.57%	42.00% ± 6.78%	98.96% ± 0.53%	63.00% ± 5.55%
70	1.80% ± 0.57%	46.00% ± 3.16%	98.58% ± 0.50%	61.33% ± 2.09%
75	2.30% ± 1.20%	46.40% ± 5.18%	97.83% ± 1.35%	58.33% ± 5.62%
80	1.70% ± 0.84%	46.40% ± 3.58%	98.50% ± 0.95%	58.00% ± 2.80%
85	1.60% ± 0.82%	46.00% ± 3.74%	98.54% ± 0.69%	59.50% ± 4.70%
<b>90</b>	<b>1.20% ± 0.76%</b>	<b>37.60% ± 6.07%</b>	<b>99.17% ± 0.63%</b>	<b>68.00% ± 6.52%</b>
95	2.90% ± 4.57%	42.00% ± 6.16%	97.38% ± 4.53%	63.00% ± 6.28%
100	1.60% ± 0.65%	41.20% ± 4.60%	98.96% ± 0.39%	63.67% ± 2.92%