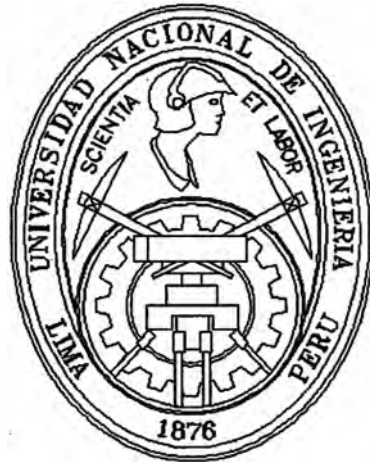


# Universidad Nacional de Ingeniería

FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS



## **Desarrollo de un Modelo basado en Redes Neuronales, para el Reconocimiento Automático de Dígitos**

### **TESIS**

Para Optar el Título Profesional de:

**INGENIERO DE SISTEMAS**

**JOSE ANTONIO VICUÑA MUÑOZ**

Lima - Perú  
1997

**Desarrollo de un modelo basado en Redes Neuronales, para el  
reconocimiento automático de dígitos**

## **Dedicatoria**

A mis padres y hermanos, cuyo apoyo incondicional permitió llevar a buen término el presente trabajo.

A todas las personas que contribuyeron enormemente con sus críticas, aportes y sugerencias.

## **Sumario**

La presente Tesis aborda el problema del reconocimiento de dígitos manuscritos de una manera automática, para lo cual empleamos la tecnología de las Redes Neuronales Artificiales. Los resultados obtenidos evidencian la posibilidad de una implementación capaz de resolver el problema planteado estando la solución no totalmente generalizada pues como se podrá suponer este problema todavía pertenece a la categoría de problemas de solución compleja.

## **Descriptores Temáticos**

Reconocimiento de patrones

Optical Character Recognition (OCR)

Redes Neuronales Artificiales

Procesamiento de Imágenes

Perceptron

Mapa Auto-Organizativo de Kohonen

Clustering

Toma de Decisiones Automática

## **Capítulo 1 Presentación**

### **1.1 Introducción**

El objetivo de la tesis es el diseño de un clasificador automático de dígitos escritos a mano.

### **1.2 Objetivos**

#### **valor por el proceso de modelamiento**

Hacemos uso de la tecnología de redes neuronales artificiales, que actualmente está brindando buenos resultados en aplicaciones de la vida real donde técnicas tradicionales han fracasado; la aplicación a nuestro problema específico nos permite evaluar las posibilidades que ofrece el mismo.

La forma de implementación presentada es general y constituye un buen marco de referencia para implementaciones futuras, es decir, no solamente es válido para el problema particular que estamos presentando.

El principal obstáculo a salvar para desarrollar aplicaciones basadas en redes neuronales artificiales lo constituye la forma como uno desarrolla la implementación, es decir, el modelo y la forma de entrenamiento ya están planteados, lo que queda por determinar es la forma como se capturan los datos del medio ambiente, su transformación y posterior introducción al modelo.

#### **valor por la aplicación sugerida**

Desde hace tiempo se vienen investigando nuevas técnicas para la captura automática de datos, dicho proceso constituye el punto de partida en la implementación de los modernos sistemas de información. Esta necesidad se basa en la búsqueda de métodos más eficientes para la captura de datos pues los que actualmente son usados dependen mucho del factor humano y la complejidad aumenta a medida que se incrementa el número de datos, una característica que identifica a los nuevos sistemas de información.

Se trata de conjugar entre eficiencia e interfase agradable para el usuario, el cual se muestra renuente a ajustarse a los patrones rígidos exigidos por los sistemas actuales, se trata pues de rescatar la forma natural que tienen los seres humanos al momento de generar su información.

El desarrollo de los sistemas de información modernos nos obliga a integrar los denominados multimedios como son: datos, texto, voz, video e imágenes, por lo que es necesario el desarrollo de técnicas que los soporten, una de las principales técnicas que cobró vigencia a raíz de estas nuevas necesidades es el reconocimiento de patrones.

### **1.3 Antecedentes**

El problema del reconocimiento de dígitos no es nuevo, podríamos pensar que la necesidad de automatizar este proceso empieza el año 1,933 con el denominado “feriado bancario” que trajo consigo una revolución en la actividad bancaria; dicha revolución no fue causada por la introducción de las computadoras en el negocio bancario (fue el instrumento para su aceleración) sino por un reconocimiento de las nuevas necesidades del mercado: las necesidades de los individuos antes que las necesidades de los negocios y de las grandes corporaciones. Dicha revolución trajo consigo un incremento en el número de cuentas corrientes de personas naturales, sistemas de créditos, tarjetas de crédito y diferentes modalidades de ahorro según las tasas de interés. Esta revolución también trajo consigo el problema de la “explosión del papel”. A finales de los años 50 la proliferación de documentos que procesar trajo

consigo un enorme problema operativo a las instituciones bancarias, es así que las computadoras digitales surgen como una alternativa para hacer frente al problema suscitado. Estas permitieron a los bancos el proceso de grandes volúmenes de datos a una gran velocidad y a un costo razonable lo que trajo consigo mejores oportunidades para continuar con el crecimiento explosivo en las operaciones bancarias. De aquí que las computadoras se convirtieron en una poderosa herramienta para la administración bancaria. Dicho crecimiento y un incremento en las utilidades de los bancos les permitió seguir buscando nuevos mercados y más clientes que atender lo cual incrementó el crecimiento; es así como las computadoras aceleraron la revolución bancaria.

Pero no sólo las instituciones bancarias se vieron beneficiadas y a la vez entorpecidas con es revolución; otras instituciones como las gubernamentales sufrieron los mismos problemas: el uso de las computadoras trajo la necesidad del almacenamiento y uso de una mayor cantidad de información, lo que significó el desarrollo de hardware y software que provean de mayor velocidad de proceso, desarrollo de sistemas con facilidades de almacenamiento y recuperación de información y sobre todo de sistemas de reconocimiento de patrones.

Actualmente estamos en la era en la cual el hombre y sus sistemas de información forman una gran alianza a fin de llevar a cabo actividades nunca antes imaginadas. Esta nueva alianza hizo de a las sociedades más productivas y elevó la calidad de vida de las personas.

Uno de los mayores problemas en el diseño de sistemas de información totalmente automáticos es el reconocimiento automático de patrones, el cual ha sido un área de estudio e investigación de muchos grupos. En estos grupos de investigación estuvieron: ingenieros, expertos en ciencias de la computación, analistas de sistemas, físicos, estadísticos, psicólogos, biólogos, fisiólogos, médicos y lingüistas. Cada grupo tratando de resolver ciertos aspectos del problema.

El problema es latente, la cantidad de formas con diferentes contenidos de información; la solución práctica se dió: la digitación de toda esa información para su posterior uso que trajo consigo nuevos problemas por resolver pero que a fin de

cuentas era una solución. Desde este punto de vista no tenemos sistemas de información totalmente automáticos.

Queda establecido por tanto la necesidad de desarrollar sistemas automáticos de reconocimiento de patrones como una condición para que en el futuro podamos tener sistemas de información totalmente automáticos.

## **1.4 Alcances**

### **1.4.1 Relación con otros trabajos desarrollados**

La aplicación de la tecnología de las Redes Neuronales Artificiales al problema del reconocimiento automático de dígitos no es nuevo, en la actualidad existen muchas implementaciones en el mercado, sin embargo, la mayor parte de ellas todavía están en “el estado del arte” y no todas ofrecen una solución satisfactoria al problema.

Entre los ejemplos más importantes tenemos los trabajos de [1] y [2].

Uno de los trabajos más populares es el mencionado en [1], donde se aplica el modelo de red neuronal conocido como “Backpropagation” implementado por AT&T y que actualmente es utilizado por el servicio postal norteamericano, se utilizó un robot en la implementación del modelo el cual lleva a cabo la tarea de reconocimiento y posterior clasificación de las cartas.

Durante el proceso de entrenamiento se utilizaron dos bases de datos: una del servicio postal norteamericano y otro del europeo.

Existen diferentes trabajos al respecto, cada uno de ellos intenta resolver el problema desde una perspectiva diferente, las principales diferencias se dan en: el modelo de red neuronal artificial a usar, el algoritmo de aprendizaje y la forma como se capturan los datos del medio ambiente. Hemos elegido el modelo de red neuronal artificial conocido como Mapa Auto-Organizativo de Kohonen (Self-Organizing Feature Map SOFM); no hemos podido encontrar un trabajo previo para el reconocimiento de dígitos que haga uso de este



modelo; el algoritmo de aprendizaje es el propuesto por el Finlandés Teuvo Kohonen. Dicho algoritmo es para el tipo de aprendizaje no supervisado y su uso está íntimamente ligado al modelo elegido. En cuanto a la forma como se capturan los datos del medio ambiente, hacemos uso de un preprocesamiento basado en software para la extracción de características que alimentarán el modelo. Para tal fin se tuvo que codificar los programas que se encargarían de llevar a cabo esta tarea. Como nota aparte queremos mencionar que las referencias en castellano son muy escasas o sencillamente no existen.

#### **1.4.2 Delimitación misma del trabajo**

Un trabajo general consiste en el reconocimiento de todos los símbolos que normalmente usamos para transmitir información; el presente trabajo cubre solamente el reconocimiento de dígitos con lo cual estamos cubriendo una buena cantidad de aplicaciones que podrían ser implementadas. Se han codificado la mayor parte de los programas utilizados para el desarrollo de nuestra implementación, lo que nos permite mayor control al momento de integrar esta solución a un sistema integral, por ejemplo, un sistema de información cuyos datos de entrada al proceso de datos sean dígitos manuscritos.

#### **1.5 Importancia del trabajo**

El desarrollo de este trabajo está orientado al reconocimiento automático de patrones, con lo cual se pretende llenar el vacío que actualmente existe para llevar a cabo la implementación de sistemas de información totalmente automáticos.

Adicionalmente se muestra los pasos a seguir para utilizar la tecnología de las redes neuronales artificiales a problemas donde otras técnicas han obtenido resultados parciales o simplemente no los han podido resolver.

El desarrollo de sistemas de reconocimiento automático de patrones había sido relegado durante bastante tiempo por la falta de técnicas adecuadas que resuelvan el problema de manera integral; con el desarrollo de las técnicas de redes neuronales artificiales y otras complementarias como Fuzzy Logic (Lógica Difusa) y Genetic Algorithms (algoritmos genéticos), se están obteniendo resultados que nos obligan a pensar como módulos necesarios en las futuras implementaciones de sistemas.

Mediante este trabajo pretendemos que más personas en nuestro país se interesen en el estudio de las redes neuronales artificiales, así como en la implementación de aplicaciones que tan buenos resultados están produciendo en otros países. Es un campo de investigación relativamente nuevo en nuestro medio y también a nivel mundial y se presenta como una alternativa de desarrollo que ofrece grandes expectativas y un punto de inicio en la forma de desarrollo que tienen los nuevos sistemas de información.

Este tipo de implementaciones ya se encuentran disponibles en el mercado pero todavía en el “estado del arte”;

## **1.6 Estructura del trabajo.**

El presente trabajo está basado en el uso de las redes neuronales artificiales, por lo que su desarrollo está íntimamente ligado a las consideraciones y sugerencias de las personas que desarrollaron los modelos. En el Capítulo 1 presentamos los objetivos de la tesis, los antecedentes y la relación con otros trabajos; la definición del problema es presentada en el Capítulo 2, así como el alcance de la solución. En el Capítulo 3 presentamos la introducción a la teoría del reconocimiento de patrones y las redes neuronales artificiales, asimismo una descripción del esquema general de solución propuesto; hemos dejado el capítulo 4 para la explicación del diseño de la interfase que brindará los datos al modelo; en el capítulo 5 se presenta la estrategia seguida para el entrenamiento de los modelos presentados y en el Capítulo 6 se hace una descripción de los componentes y la forma de implementación de un clasificador postal.

## **Capítulo 2 Definición del problema**

### **2.1 Definición del objeto de estudio**

Estamos interesados en mostrar la forma como implementar un sistema que nos permita llevar a cabo el reconocimiento automático de dígitos manuscritos. Dicha implementación supone que previamente se han tenido que resolver diversos problemas; la forma de solución planteada para cada uno de ellos es expuesta a lo largo de la tesis. Nuestro interés radica en la presentación del modelo que efectuará la tarea del reconocimiento pues constituye la pieza fundamental de este sistema.

### **2.2 Definición del ámbito de estudio**

El problema del reconocimiento es bastante genérico en relación al contexto donde se puede encontrar el dígito:

Encontramos los dígitos en formatos preimpresos, los cuales son de mucha ayuda al momento de escribirlos pero dificultan la tarea del reconocimiento automático.

No encontramos a los dígitos de manera aislada sino que aparecen en grupos y muchas veces combinados con caracteres e imágenes.

Los dígitos son escritos en diferentes tamaños y grosores, con diversos tipos de tinta o lápiz.

Los dígitos son ilegibles por la presencia de elementos extraños que tienden a confundir inclusive a los seres humanos.

La técnica que describimos aborda en gran medida todos estos problemas, sin embargo y para efectos prácticos es necesario tener ciertas restricciones, las cuales son descritas más adelante, que lejos de restarle generalidad a la implementación nos permite presentar una solución que aunque práctica resulta muy útil en los problemas de clasificación.

### **2.3 Formulación de pregunta eje**

El sistema debe ser capaz de llevar a cabo el reconocimiento de los dígitos para posteriormente llevar a cabo tareas tales como la clasificación y ordenamiento automático, existen muchas aplicaciones que demandan mucho tiempo en la captura de la información y la validación de la misma, pues son tareas efectuadas de manera manual y con el consiguiente margen de error propio de la operación, el sistema alternativo, si bien es cierto inicialmente no nos puede brindar un procedimiento libre de errores pueden llevar a cabo la misma tarea de una manera más rápida y sobre todo con la capacidad de ajustar el margen de error pues son sistemas que necesariamente necesitan de la intervención humana para resolver las situaciones que ellos no pueden resolver.

### **2.4 Diferencias con los OCR**

Los OCR (Optical Character Recognition) son sistemas que llevan a cabo la misma labor, sin embargo presentan algunas diferencias que son presentadas a continuación:

Por la forma de implementación de los OCR, no es posible utilizarlos para llevar a cabo un proceso de clasificación pues lo que hacen es la interpretación global de un texto previamente scaneado y llevarlo a un procesador de texto. Si bien es cierto existe un proceso intermedio de reconocimiento de caracteres no es el objetivo principal llegar a determinar de que o quienes se trata.

Existen formas de utilizar estos OCR para las labores de clasificación pero la interfase no sería natural ni adecuada.

La mayor parte de los OCR trabajan sobre tipografía preimpresa, recién están apareciendo en el mercado OCR's que también llevan a cabo el reconocimiento de caracteres manuscritos, pero están en el "estado del arte".

## **Capítulo 3 Marco Teórico**

### **3.1 Reconocimiento de patrones**

#### **3.1.1 Conceptos básicos referidos al reconocimiento de patrones**

##### **Reconocimiento de Patrones**

La categorización de un dato de entrada a una clase identificada previamente mediante la extracción de características o atributos del dato a partir de los detalles que se consideren relevantes.

##### **Clase patrón**

Es una categoría determinada por atributos comunes que deben mantener los miembros de la misma. Un patrón es la descripción de algún miembro de la categoría representando a la clase patrón.

##### **Reconocimiento**

El reconocimiento es entendido como un atributo básico del comportamiento humano y de otros organismos vivientes. Un patrón es la descripción de un objeto. Estamos llevando a cabo actos de reconocimiento en cada instante de nuestras vidas. Reconocemos los objetos que nos rodean, nos movemos y actuamos en relación a ellos. Por ejemplo podemos ver a un amigo en una multitud y reconocer lo que dice, podemos reconocer a un individuo por su voz, podemos leer manuscritos, podemos reconocer risas de molestias.

De acuerdo a la naturaleza de los patrones a ser reconocidos, podemos dividir nuestros actos de reconocimiento en dos tipos: el reconocimiento concreto y el reconocimiento abstracto; reconocemos caracteres, pintura, música y los objetos que nos rodean; de otro lado podemos reconocer argumentos en los cuales está basado una película o la solución a un problema.

Cuando una persona percibe un patrón, hace una inferencia inductiva y asocia su percepción con algunos conceptos generales que ha ido derivando en función a su experiencia pasada.

El reconocimiento llevado a cabo por los seres humanos es en realidad una cuestión de estimación a fin de asociar el patrón a alguna de las poblaciones determinadas de antemano en un proceso previo de reconocimiento y clasificación de tales poblaciones; de aquí que el problema del reconocimiento puede ser considerado como una asociación no con patrones individuales sino entre poblaciones, vía la búsqueda de características o atributos invariables de los miembros de la población.

El estudio del problema del reconocimiento de patrones puede ser dividido lógicamente en dos categorías:

- 1ro El estudio de la capacidad de los seres humanos y otras formas de vida del reconocimiento de patrones.
- 2do El desarrollo de teorías y técnicas para el diseño de sistemas que lleven a cabo la tarea del reconocimiento de patrones para una aplicación específica.

La primera categoría es objeto de estudio por disciplinas tales como psicología, fisiología y biología; la segunda categoría es estudiada por las diversas ramas de ingeniería, ciencias de la computación y ciencias de la información.

### **3.1.2 Conceptos de diseño y metodologías**

Los conceptos de diseño para el reconocimiento automático de patrones está motivado por la forma como las clases patrón con caracterizadas y definidas. Existen muchas posibilidades:

#### 1.- Clasificación exacta

El reconocimiento automático de patrones es llevado a cabo mediante un matching contra los patrones ya definidos y almacenados de antemano. El conjunto de patrones de una misma clase son almacenados en el sistema de reconocimiento de patrones. Cuando se presenta un patrón desconocido, éste es comparado uno a uno con los patrones almacenados previamente. Por ejemplo: si letras de diferentes formas y tamaños son almacenados en un sistema de reconocimiento de patrones, una letra desconocida podrá ser reconocida sobre la base de una clasificación exacta siempre y cuando no haya sufrido distorsiones al momento de obtenerlo, tinta no legible, el tipo de papel u otros. Esto constituye un método bastante simple pero importante para para implementar un sistema de reconocimiento de patrones relativamente barato y útil para determinados propósitos. El éxito de las implementaciones basadas en este modelo radica en que los patrones desconocidos sean copias bastante parecidas de los patrones almacenados previamente. Los primeros sistemas OCR (Optical Character Recognition) siguen esta implementación.

#### 2.- Propiedad Común

La caracterización de clases patrón por propiedades comunes compartidas por todos sus miembros nos sugiere el reconocimiento automático mediante la detección y reconocimiento de características similares. El concepto básico de este modelo es que el patrón que pertenece a una clase posee ciertas propiedades comunes o atributos que refleja similitud entre



todos los componentes. Estas características comunes pueden ser almacenados en el sistema de reconocimiento automático de patrones. Cuando un patrón desconocido ingresa al sistema, se obtienen sus características o atributos y luego se comparan contra las características almacenadas. El esquema de reconocimiento clasificará el patrón desconocido a la clase patrón con características similares. El principal problema en este tipo de implementaciones es la determinación de características comunes de un conjunto finito de patrones ejemplo conocidos de antemano.

Este modelo es superior al presentado anteriormente en muchos aspectos: requiere de menos espacio para almacenar la información relevante, las características de la clase patrón permanecen invariantes al extremo que puede reconocer patrones con algunas variaciones (la potencia del modelo está en función a cuanta variación en los patrones puede soportar), las características de la clase patrón pueden ser determinadas tomando como base patrones ejemplo representativos. Por otro lado es extremadamente difícil, sino imposible la determinación de las características más relevantes que permitan un discriminación entre las diferentes clases patrón; para implementar este modelo es necesario el desarrollo de técnicas adecuadas de extracción de características en función a la aplicación a desarrollar.

### 3.- Clustering

Cuando los patrones de una clase son vectores cuyos componentes son números reales, una clase patrón puede ser caracterizada por sus propiedades de clustering dentro de una espacio patrón. El diseño de un sistema de reconocimiento de patrones está en función al arreglo geométrico de los diferentes clusters patrón. Si las clases son

caracterizadas por clusters bastante diferenciados, es factible esquemas de reconocimiento sencillos tales como el concepto de la distancia mínima; si los clusters se traslapan es necesario el uso de técnicas sofisticadas para la partición del espacio patrón. El traslape de los clusters puede ser el resultado de una deficiencia en la información observada o la presencia de ruido en el momento de la medición, es posible una minimización del traslape incrementando el número y la calidad de las mediciones de los patrones de una clase.

Los conceptos básicos de diseño para el reconocimiento automático de patrones descritos anteriormente, pueden ser implementados por tres principales metodologías: heurísticas, matemáticas y lingüísticas (sintácticas). Es común encontrar una combinación de estas metodologías en los sistemas de reconocimiento automático de patrones.

#### 1.- Métodos heurísticos

La aproximación heurística está basada en la intuición y experiencia de la persona que lo lleva a cabo, haciendo uso de los conceptos de clasificación exacta y propiedades comunes. El diseño de un sistema que sigue este principio consiste en el desarrollo de procedimientos predeterminados para la tarea a llevar a cabo. Aunque la aproximación heurística es ampliamente utilizada en el diseño de los sistemas de reconocimiento automático de patrones, poco se puede decir acerca de la generalización de sus principios desde que cada problema requiere la aplicación de reglas de diseño desarrolladas a la medida de la aplicación; de todo esto se deduce que la estructura y performance de un sistema heurístico depende en alto grado de la intuición y experiencia de los diseñadores del sistema.

#### 2.- Métodos matemáticos

La aproximación matemática está basada en reglas de clasificación las cuales son formuladas y derivadas dentro de una estructura matemática, haciendo uso de los conceptos de propiedad común y clustering. Este método es en contraste con la aproximación heurística en la cual las decisiones están basadas en reglas establecidas según la aplicación a desarrollar.

La aproximación matemática puede ser dividida en dos categorías: determinística y estadística.

La aproximación determinística está basada en una estructura matemática que no emplea de manera explícita las propiedades estadísticas de las clases patrón bajo consideración.

La aproximación estadística está basada en reglas de clasificación matemática, las cuales son estructuradas y derivadas de una estructura estadística. El diseño de un clasificador de patrones estadístico está basado en las reglas de clasificación de Bayes y sus variantes; esta regla nos dice que tenemos un clasificador óptimo cuando la función de densidad de probabilidad de cada población patrón y la probabilidad de ocurrencia de cada clase patrón son conocidas.

### 3.- Métodos lingüísticos (sintácticos)

La caracterización de patrones por elementos primitivos (subpatrones) y sus relaciones, sugiere el desarrollo de sistemas de reconocimiento automático de patrones mediante la aproximación lingüística, haciendo uso del concepto de propiedad común. Un patrón puede ser descrito por una estructura jerárquica de subpatrones análogo a la estructura sintáctica de los lenguajes. Esto permite la aplicación de la teoría de los lenguajes

formales al problema del reconocimiento de patrones. Un patrón gramático es considerado como un conjunto finito de elementos. La esencia de esta aproximación radica en la selección de patrones primitivos, la forma como se estructuran para unirse y las relaciones en los patrones gramaticales. Esta aproximación es útil para los patrones que no pueden ser numéricamente medidos.

## **3.2 Redes Neuronales**

### **3.2.1 Introducción**

Las Redes Neuronales y el procesamiento de señales usadas en las diferentes aplicaciones de la vida real, pueden ser divididas en tres categorías:

#### Feedforward Networks:

Transforman el conjunto de señales de entradas en un conjunto de señales de salida; la transformación entrada/salida deseada es determinada por factores externos, mediante un ajuste supervisado de los parámetros del sistema.

#### Feedback Networks:

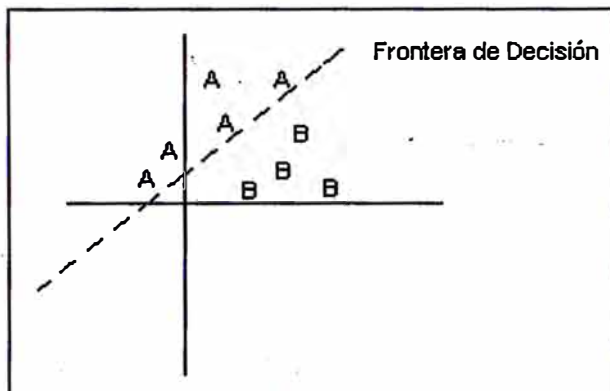
La información de entrada define el estado de actividad inicial de un sistema retroalimentado y después de diferentes transiciones el estado final es identificado mediante parámetros de estabilización establecidos generalmente por una medida de error.

#### Self-Organizing Networks:

Células vecinas en una red neuronal compiten en sus actividades mediante una interacción lateral mutua y mediante el desarrollo adaptativo se convierten en detectores específicos de diferentes patrones de señales.

### 3.2.2 Teoría del Perceptrón

Esta red neuronal es la más sencilla pero genera mucho interés entre los investigadores que recién se inician en esta área por su habilidad para reconocer patrones simples. Tomemos el ejemplo de reconocer una entrada desconocida pero perteneciente a uno de dos clases denotadas por A y B tal como se muestra en la Figura 3.1



**Figura 3.1 Problema con dos Clases Patrón**

La única neurona del modelo calcula la suma ponderada de los pesos  $y_i$  los valores de las entradas; el resultado es transformado mediante una función signo a fin de obtener sólo dos valores: +1 ó -1. La regla de decisión está basada en la regla:

la entrada desconocida pertenece a la clase A si el valor obtenido es +1 o pertenece a la clase B si el valor obtenido es -1.

Una técnica muy útil para analizar el comportamiento de la red neuronal es dibujar en un plano cartesiano la ubicación de los elementos pertenecientes a cada clase; esto nos permitirá establecer si es posible obtener una frontera de decisión.

Los pesos de las conexiones y el umbral ( $w_0$ ) pueden ser fijos a adaptados utilizando para ello una serie de algoritmos de aprendizaje, presentamos aquí el algoritmo desarrollado por Rosembat.

### Algoritmo

<b>Paso 1</b>	<b>Inicializar pesos (<math>w_1..w_N</math>) y el umbral (<math>w_0</math>)</b> La inicialización se efectúa mediante la asignación de pequeños números aleatorios.
<b>Paso 2</b>	<b>Presentar Entrada Desconocida y Salida Deseada</b> Presentar un conjunto de valores $x_i$ y la salida deseada $d$
<b>Paso 3</b>	<b>Calcular la Salida Obtenida</b> $z =$ Suma ponderada de los $w_i$ y los $x_i$ $o =$ Función signo( $z$ )
<b>Paso 4</b>	<b>Adaptar los Pesos <math>w_i</math></b> $w_i(t+1) = w_i(t) + \text{etha} (d_t - o_t) x_i(t)$ $d = +1$ si la entrada pertenece a la Clase A $d = -1$ si la entrada pertenece a la Clase B etha: coeficiente de aprendizaje
<b>Paso 5</b>	<b>Repetir desde el paso 2</b> Mientras no se haya concluído con el tiempo asignado al aprendizaje

Rosembat determinó que si las muestras utilizadas para el entrenamiento permitían el trazado de una frontera de decisión lineal, entonces el modelo iba a

converger, de esta forma podía ser utilizado para el reconocimiento de patrones pertenecientes a las clases a partir de las cuales fue entrenado.

### **3.2.3 Teoría del SOFM**

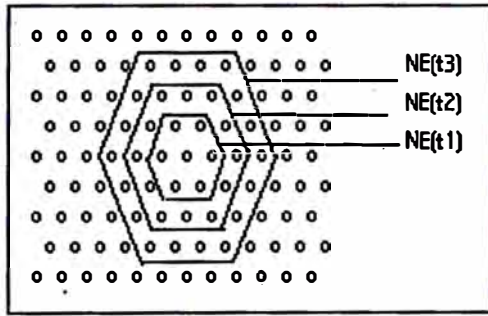
Hace mas de 100 años una detallada descripción topográfica del cerebro, especialmente de la corteza cerebral podía ser deducida a partir de la deficiencia funcional y ciertos impedimentos inducidos por lesiones de distinto tipo tales como hemorragias, tumores y malformaciones. Diferentes regiones de la corteza cerebral parecían estar dedicadas a tareas específicas.

Kohonen presentó un algoritmo al cual denominó Self-Organizing Feature Map (SOFM) que reproduce el comportamiento en la corteza cerebral anteriormente descrito.

El algoritmo de Kohonen básicamente ajusta los pesos de las conexiones entre las neuronas de entrada y las neuronas de salida dispuestos estos últimos en un arreglo dimensional tal como se muestra en la Figura 3.2 al cual también se le denomina Mapa de Características o Mapa Topográfico. Las neuronas o nodos de salida se encuentran extensivamente interconectadas con muchas conexiones locales.

Valores continuos en la forma de vectores son presentados a través de las neuronas de entrada en forma secuencial en el tiempo y sin especificar la salida deseada.

Después que los vectores de entrada han sido presentadas a la red neuronal un número suficiente de veces, los pesos de las conexiones almacenarán las características dispuestas a manera de sub-espacios o clusters de los vectores de características; en otras palabras, los pesos de la conexiones estarán dispuestos de manera topográfica en la cual las neuronas o nodos cercanos serán sensitivos o reconocerán las entradas que sean similares.



**Figura 3.2 Vecindades Topológicas**

$$t_3 < t_2 < t_1$$

El algoritmo que forma el Mapa de Características requiere que se defina la vecindad de cada neurona; dicha vecindad decrece en el tiempo durante el entrenamiento.

Los pesos de las conexiones son inicializados con valores aleatorios pequeños lo cual nos asegura la condición que al inicio del entrenamiento el conocimiento de la red neuronal acerca de lo que va a reconocer es CERO. El algoritmo propuesto por Kohonen es presentado en el siguiente cuadro:

**Algoritmo**

<b>Paso</b> <b>1</b>	<b>Inicializar los pesos <math>w_0</math></b> Inicializar los pesos de N entradas y M salidas con pequeños valores aleatorios. Inicializar el radio inicial de vecindad.
<b>Paso</b> <b>2</b>	<b>Presentar una nueva entrada</b>
<b>Paso</b> <b>3</b>	<b>Calcular la distancia a todos las neuronas</b> Calcular la distancia $d_j$ entre la entrada y cada neurona de salida $j$ usando: $d_j = \text{Suma ponderada } (x_i(t) - w_{ij}(t))^2$



	donde $x_i(t)$ es la entrada por la neurona $j$ en el tiempo $t$ y $w_{ij}(t)$ es el peso entre la neurona de entrada $i$ y la neurona de salida $j$ en el tiempo $t$
<b>Paso 4</b>	<b>Seleccionar la neurona de salida con distancia mínima</b> Seleccionar la neurona $j^*$ como la neurona de salida con la menor distancia $d_j$
<b>Paso 5</b>	<b>Actualizar los pesos de la neurona <math>j^*</math> y las neuronas vecinas</b> Los pesos son actualizados para la neurona $j^*$ y todas las neuronas de la vecindad definida por $Ne_{j^*}(t)$ . Los nuevos pesos serán: $w_{ij}(t + 1) = w_{ij}(t) + \text{etha}(t) (x_i(t) - w_{ij}(t))$ para todos los $j$ que pertenecen a $Ne_{j^*}(t)$ etha: coeficiente de aprendizaje que decrece en el tiempo.
<b>Paso 6</b>	<b>Repetir desde el Paso 2</b> Mientras no haya concluido el tiempo asignado para el aprendizaje

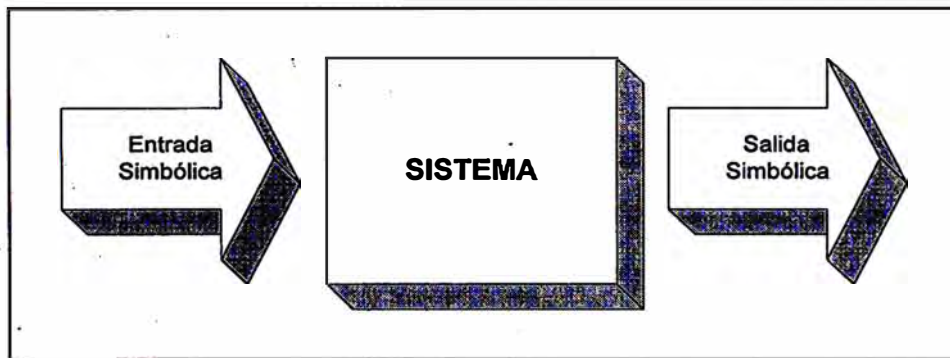
El vector de características es la salida simbólica de la Interfase, y se denomina vector porque lo que se obtiene es un conjunto de valores numéricos que a nuestro juicio representan las características de los dígitos. La potencia de las redes neuronales radica en que el conocimiento que almacenan no son los dígitos en sí sino sus características, razón por la cual pueden inferir al momento de efectuar una clasificación.

Se dice que los seres humanos llevamos a cabo procesos de reconocimiento porque almacenamos los patrones característicos; en el caso de los dígitos nosotros no aprendemos sobre la base de almacenar las imágenes de todos los dígitos sino que de alguna forma somos entrenados para almacenar sus formas, lo que luego nos permite reconocer dígitos que nunca hayamos visto; ésta es la

característica que este modelo incorpora, razón por la cual fue elegida para la solución de nuestro modelo.

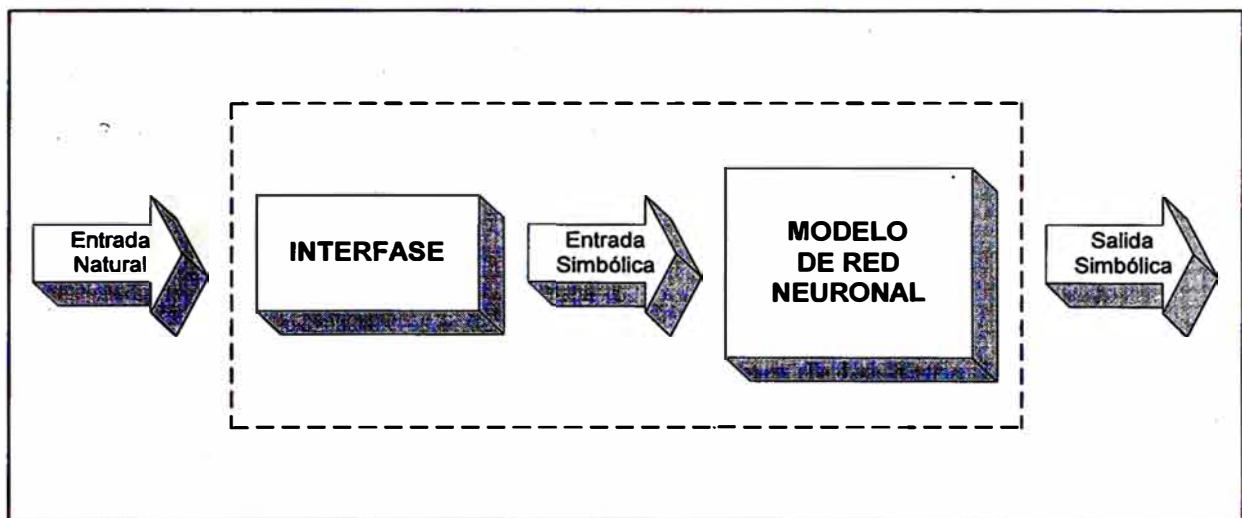
### 3.3 Esquema general de la implementación desarrollada

La Figura 3.3 nos muestra un esquema general de implementación basado en computadoras digitales; tanto la entrada como la salida son simbólicas y el diseño del sistema es una implementación algorítmica para computadoras de uno o más procesadores. Las entradas se tienen que preparar a partir de la información del medio ambiente.



**Figura 3.3 Esquema tradicional de procesamiento**

En la Figura 3.4 mostramos el esquema general que debería tener nuestra implementación, para tal fin es necesario el uso de una neurocomputadora que en teoría es capaz de procesar los datos tomados directamente del medio ambiente. Como se puede apreciar la entrada al sistema es natural (multimedios) lo cual evita la “transcripción” de los datos, es decir, tomados de la fuentes donde se originan, lo que significa una minimización en los errores producto de los procesos de digitación. Como ventaja adicional mencionamos que el procesamiento en paralelo es natural en este tipo de computadoras lo que permite una implementación real de las soluciones basadas en Redes Neuronales Artificiales.



**Figura 3.4 Implementación sugerida**

Ante la imposibilidad de contar con una neurocomputadora hemos tenido que pensar en una implementación teniendo en cuenta que sólo contamos con computadoras digitales, para lo cual tenemos que recurrir al desarrollo de una interfase que se encargue de transformar las entradas naturales en entradas simbólicas a nuestro sistema basado en Redes Neuronales Artificiales.

Asimismo se ha tenido que plantear la implementación del modelo de red neuronal pensando en que se ejecutará en una computadora digital.

Este modelo es utilizado en la fase de entrenamiento del modelo y constituye la implementación final del mismo sin ninguna variante para el proceso de clasificación.

Cabe mencionar que la salida obtenida es simbólica; de esta forma es posible utilizar las salidas del modelo en otros módulos de aplicaciones ya desarrolladas y que se ejecutan en computadoras digitales; ésta es la característica de los modernos sistemas de procesamiento de datos.

### 3.3.1 Algoritmo para determinación de solución

A continuación presentamos el algoritmo general que nos permitirá encontrar la solución al problema planteado:

#### Pasos:

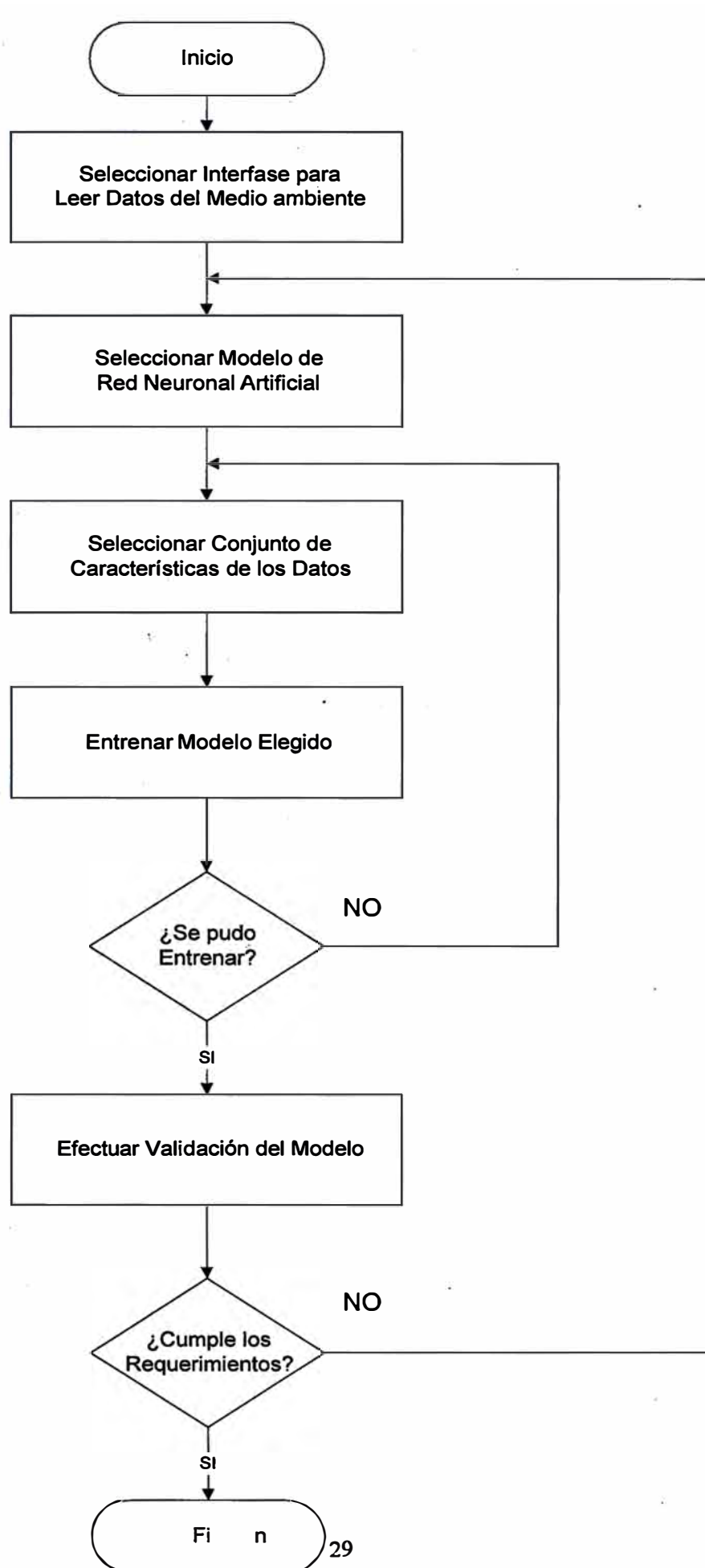
- 1 Determinar la interfase a utilizar para la lectura de los datos desde el medio ambiente (entrada natural).
- 2 Seleccionar modelo
- 3 Seleccionar el conjunto de características
- 4 Entrenar al modelo de Red Neuronal  
Si el modelo no se puede entrenar o no converge regresar al punto 3
- 5 Validar el modelo  
Si el modelo no satisface requerimientos regresar al punto 2
- 6 Modelo listo para ser utilizado

#### Explicación del algoritmo:

- 1 Aquí se tiene que resolver el problema de la lectura de los datos. Esta etapa es necesaria pues la implementación del modelo es en una computadora digital; en el capítulo 4 se describe el diseño de la interfase.
- 2 Hemos elegido dos de los modelos más conocidos: El Perceptron y el SOFM; el primero porque permite una aproximación práctica muy importante para el entendimiento de la técnica y el segundo por ser la solución elegida. Una explicación detallada se realiza en el capítulo 5
- 3 Está en función a los modelos elegidos y la explicación de los mismos se realiza en el capítulo 5.

- 4 Las consideraciones para el entrenamiento de cada modelo se encuentran especificadas en el capítulo 5
  
- 5 La validación del modelo se realiza en el capítulo 5

La Figura 3.5 nos muestra el diagrama de flujo del algoritmo



## **Capítulo 4 Diseño de la Interfase**

Este capítulo consta de dos partes: la primera es el resultado del análisis para determinar los problemas a resolver para este tipo de implementaciones; en la segunda parte se presentan los requerimientos que debe cumplir la interfase para cada uno de los modelos elegidos así como las interfases seleccionadas. Posteriormente se describe la interfase desarrollada para la captura de datos para el modelo Perceptron y la interfase para el modelo SOFM.

### **4.1 Problemas por resolver para la implementación**

A continuación se describen los principales inconvenientes encontrados en el diseño de la interfase que nos permita la captura de los datos del medio ambiente (entrada natural).

#### **4.1.1 En la escritura de los números**

##### Dígitos dentro de un contexto

Los dígitos se presentan de manera aislada o formando parte de un contexto tales como libros, periódicos, revistas o formas preimpresas (vouchers, sobres de cartas, etc), que dificultan la extracción de sus características; este trabajo contempla sólo a los dígitos encontrados de manera aislada. Un esquema más general requiere el uso de algoritmos que actúen de filtros para aislar los dígitos del contexto en el cual se presentan. Otra forma de solución es mediante el uso de formas preimpresas que guíen a las personas mediante espacios reservados para escritura.

### Elección de las muestras

Los modelos de redes neuronales son independientes de las datos de entrada, es decir, para un adecuado funcionamiento no requieren características especiales que deben cumplir los datos. Por lo tanto poca o ninguna atención es tomada en cuenta durante el proceso de selección de las muestras. La falta de una estandarización en los dígitos que cada persona posee no ha permitido el éxito de algoritmos basados en otras tecnologías, sin embargo como mostraremos más adelante las Redes Neuronales Artificiales reaccionan positivamente a esta falta de estandarización.

### Instrumento utilizado en la escritura

Los dígitos son escritos mediante la utilización de diferentes tipos de instrumento (lápices, lapiceros, etc), lo que contribuye a incrementar la gama de dígitos encontrados en lo que ha grosor respecta; el algoritmo de preprocesamiento presentado ha mostrado ser invariable a estas características.

### Tamaño de los dígitos

No se puede hablar de un tamaño estándar cuando nos referimos a dígitos manuscritos, está en función al espacio asignado para tal fin y según el contexto en el cual se localicen. Para nuestros propósitos optamos por “estandarizar” dicho tamaño mediante el empleo de una forma preimpresa, mediante la cual demarcamos una área para la escritura de los mismos. La única condición que impartimos fue que no desborden el área asignada para tal fin. Utilizamos papel milimetrado y el área asignada correspondió a un cuadrado de 1 cm. de lado. Los dígitos así obtenidos fueron digitalizados mediante un scanner y se procedió a obtener imágenes de 64 x 64 bits, espacio suficiente para presentar los dígitos de una manera adecuada. Según las



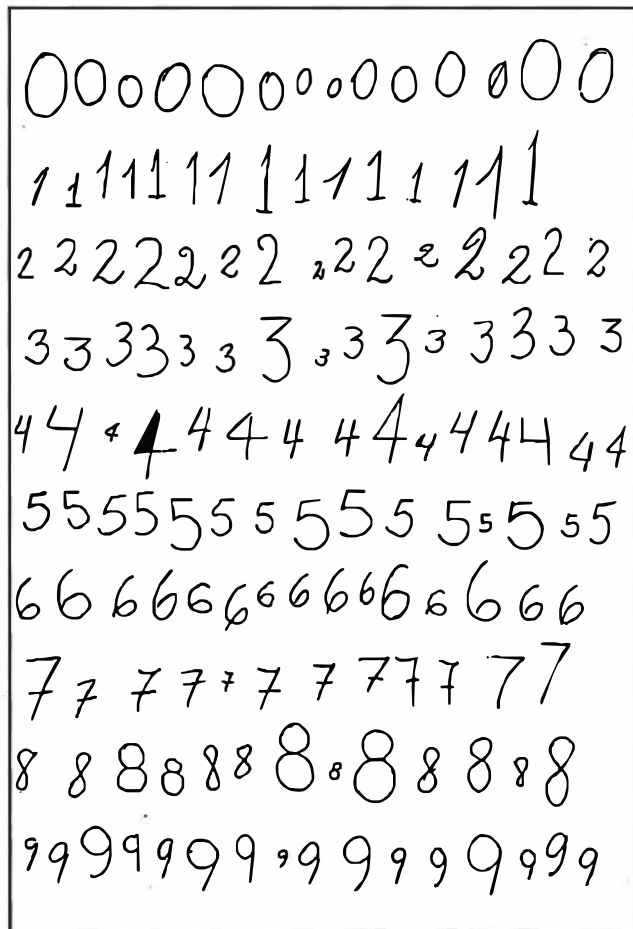
publicaciones revisadas el tamaño estandar de los dígitos tiene el tamaño descrito.

### Marcas extrañas junto con los dígitos

En la práctica encontramos marcas extrañas que se confunden con la información del dígito; los sistemas de reconocimiento de dígitos actualmente en operación incluyen técnicas sofisticadas de depuración de los mismos. Estas técnicas son conocidas como filtros y el objetivo principal es la eliminación de marcas extrañas dentro de la información. Dichos filtros ya forman parte de paquetes tales como el Image Processing Toolbox de MatLab. El uso de tales filtros está en función a las condiciones iniciales que se pueda tener. Por lo general dichas marcas son más leves que la de los dígitos, de otro modo sería difícil aún para los seres humanos distinguir el dígito escrito.

Mediante la elección de determinadas condiciones tales como un formato preimpreso libre de marcas nos permite no tener que recurrir a dichos filtros, esto podría constituirse como una condición no necesaria pero recomendable, la solución general se deja para futuras mejoras al modelo.

Las muestras de dígitos, utilizadas en la fase de entrenamiento, son mostradas en la Figura 4.1 Dichos dígitos ya pasaron por la interfase desarrollada.



**Figura 4.1 Dígitos para el entrenamiento**

#### **4.1.2 En la Digitalización de los números**

##### Elección del tamaño estándar

No se utilizan técnicas de compresión o expansión de los dígitos pues el algoritmo utilizado se muestra invariable al tamaño de los dígitos tomados como muestra. Se han tomado muestras de diferentes tamaños y formas pero dentro del tamaño predeterminado para la imagen que contendrá al dígito (64x64 pixels).

### Posición del dígito dentro de la imagen

La posición es libre dentro del cuadrado asignado para la escritura, es decir, no fue requisito que por ejemplo trataran de escribir en el centro del mismo, el algoritmo de preprocesamiento implementado es independiente de la posición del dígito dentro de la imagen. La única suposición que se hizo es que la orientación vertical de la imagen coincida con la orientación vertical que siempre se tiene al momento de escribir el dígito independiente de si el dígito es escrito de manera recta o inclinada hacia algún costado. Con esto queremos decir que los dígitos que no conserven una posición normal de escritura no serán reconocidos.

El algoritmo diseñado para la extracción de características no considera a los dígitos que hayan sufrido rotación alguna.

Para conseguir una correcta alineación del dígito es necesario adecuar el scanner a fin de que este proceso sea lo más automático posible, para ello es necesario diseñar un scanner por donde pasen las formas conteniendo los dígitos de una forma similar a las utilizadas en los procesos de etiquetado con la diferencia de no se etiquetan sino se obtiene la imagen de los dígitos que van pasando. No hemos implementado dicho procedimiento pero lo consideramos factible y un punto por resolver como mejora a este trabajo.

### Elección del formato de digitalización

Podemos elegir hasta dos procedimientos de digitalización ampliamente usados: la primera es mediante el uso de una cámara de video y la segunda es mediante el uso de un scanner.

Ambos formas de digitalización nos producen varios formatos de imágenes que luego pueden ser visualizados por software de tratamiento de imágenes.

Hemos elegido el uso del scanner no por ser más eficiente sino por la ventaja práctica de obtener formatos de imagen ampliamente conocidos, como es el caso del formato BMP (BitMaP). Una interpretación del formato BMP nos permite “leer” la información de la imagen y obtener la representación del mismo en un matriz.

Esta conversión constituyó la pieza fundamental en nuestro desarrollo, pues a partir de él fue posible la implementación de los algoritmos de extracción de características de los dígitos, punto de partida para el desarrollo de los modelos basados en Redes Neuronales Artificiales.

#### Eliminación de marcas extrañas

No se ha tenido que resolver mucho en este punto dadas las condiciones iniciales, sin embargo, es posible una ampliación de la cobertura mediante un trabajo de filtración de dichas marcas. Cabe resaltar que el uso de un nivel de resolución adecuado puede permitir la eliminación natural de dichas marcas tal y como ocurrió en nuestro caso.

#### Elección de la resolución

Aprovechamos las características de los scanners de permitirnos digitalizar imágenes hacia diferentes formatos ya sea a colores o en tonos de gris (grayscale), los dígitos fueron digitalizados a un formato de imagen grayscale de 16 tonos de gris.

Las imágenes así obtenidas tienen suficiente resolución como para visualizar todos los detalles de los dígitos.

Aquí es necesario señalar que estas consideraciones fueron utilizadas para la obtención de los vectores de características que posteriormente fueron utilizados en el entrenamiento del modelo.

## **4.2. Diseño de la Interfase para digitalizar los dígitos**

El objetivo en esta etapa es obtener el equivalente matricial de la imagen de los dígitos. Dicha información es punto de partida para el entrenamiento de nuestro modelo de red neuronal.

La interfase que aquí describimos difiere de las que normalmente se usan en las implementaciones de sistemas basados en Redes Neuronales Artificiales; la principal diferencia radica en que dichas implementaciones alimentan directamente al modelo con información recibida sin un preprocesamiento que no sea el “arreglo” de los dígitos antes de ingresar al modelo, aquí por el contrario se ha preferido realizar una implementación previa, vía software, que nos permite presentar datos simbólicos trabajados de antemano, lo cual como se explica más adelante hace que el modelo no necesite demasiadas neuronas y las implementaciones prácticas sean sencillas de implementar.

La implementación del modelo que aquí presentamos es básicamente un programa para que se ejecute en un computadora de tipo digital, entonces necesitamos que la interfase alimente a dicho programa con información simbólica, el tipo de información que estas computadoras manejan, en otras palabras necesitamos que la interfase traslade la información obtenida, una imagen del dígito por ejemplo, a un elemento simbólico como podría ser la matriz de dicha imagen.

### **4.2.1 Algoritmo propuesto**

- 1 Elegir el conjunto de muestras
- 2 Aislar los dígitos
- 3 Digitalizar el dígito en formato BMP de 16 tonos de gris
- 4 Filtrar la imagen

- 5 Convertir la imagen a BMP monocromático
- 6 Obtener el equivalente matricial de la imagen
- 7 Repetir desde el paso 2 hasta terminar con todas las muestras

## **1 Seleccionar la muestra**

Buscar dígitos escritos por diferentes personas de acuerdo a las indicaciones establecidas previamente. Es necesario dotar al modelo durante el aprendizaje de la mayor cantidad de combinaciones posibles a fin de obtener algo general.

## **2 Digitalizar el dígito en formato BMP de 16 tonos de gris**

Usar un scanner para obtener el formato solicitado. adicionalmente al formato es necesario que la imagen obtenida sea de 64 x 64 pixels tratando en lo posible de que el dígito no desborde dicho marco. Adicionalmente se asigna un nombre específico a cada imagen.

Por ejemplo D16X-YY.BMP

donde:

D16: Indica que la imagen es de 16 colores

X: 0:cero, 1:uno, 2:dos, ..., 9:nueve

YY: Secuencia de grabación

BMP: Indica formato de la imagen (gráfico)

Se trató en lo posible de obtener una imagen de 64 x 64 pixels. Es posible implementar un proceso de “cortado” automático de la imagen al tamaño deseado. No implementamos esta opción la cual dejamos para futuras mejoras a este trabajo.

### 3 Filtrar la imagen

Aún cuando esta etapa no fue implementada lo mencionamos como una mejora a la captura de datos del medio tal y como se presenten. La implementación es mediante la codificación de un programa que se encargue de dicha tarea siguiendo los algoritmos disponibles para tal fin.

### 4 Convertir la imagen a BMP monocromático

Cada pixel tiene asignado un número que indica su nivel de gris:

Número	Porcentaje de Negro (%)
0	0 (Blanco)
1	6.25
2	12.5
3	18.75
4	25
5	31.25
·	
15	93.75
16	100

#### Regla de conversión:

- a) Para cada pixel de la imagen monocromática tomar el promedio de los porcentajes de Negro del pixel y de los que lo rodean.
- b) Si Promedio  $\geq$  50 %

Entonces      Nuevo valor del pixel es Negro  
Sino            Nuevo valor del pixel es Blanco

c)      Asignar nuevo identificador.

Ejemplo: D02X-YY.BMP

donde:

D02:    Indica que la imagen es de 2 colores (monocromática)

X:      0:cero, 1:uno, 2:dos, ..., 9:nueve

YY:    secuencia de grabación

BMP:    Indica formato de la imagen (gráfico)

Para tal fin se codificó un programa en lenguaje C que permita dicha conversión de una manera automática.

## **5      Obtener el equivalente matricial de la imagen**

Leer la información de la imagen y obtener su equivalente en una matriz de 64 x 64 bytes. Almacenar un identificador al momento de grabar información resultante:

por ejemplo: MX-YY.TXT

donde:

M:      Indica que el archivo contiene el equivalente en ASCII de la imagen monocromática

X:      0:cero, 1:uno, 2:dos, ..., 9:nueve

YY:    secuencia de grabación

TXT:    Indica formato de la imagen (ASCII o texto)

Adicionalmente la información hallada en el archivo será:

1:      Negro, presencia de información perteneciente al dígito en dicha posición

0:      Blanco, el dígito no ocupa dicha posición dentro de la imagen.



Este paso no es posible si previamente no se conoce la forma como el formato BMP almacena la información acerca de la forma de ubicar cada pixel así como el color que almacena.

Se eligió el formato BMP pues se pudo obtener información acerca de como almacena la información requerida para nuestro propósito.

Posteriormente se codificó un programa en C que llevara a cabo dicha tarea.

#### **6 Repetir desde el paso 2 hasta terminar con todas las muestras**

Repetir los pasos para cada uno de las muestras seleccionadas. Con esto concluimos la tarea referente a la lectura de la información tomada del medio ambiente.

### **4.3 Elección del vector de características para entrenar al Perceptron**

Se acostumbra a llamar PREPROCESAMIENTO a esta etapa dentro del desarrollo de aplicaciones basadas en Redes Neuronales Artificiales.

#### **4.3.1 Objetivo**

Obtener el par de características que mejor describan a cada uno de los dígitos.

#### **4.3.2 Condiciones a cumplir**

El modelo Perceptron implementado nos permite una clasificación de sólo dos dígitos; hemos elegido los dígitos uno (1) y siete(7) para probar el modelo pues son dígitos que tienen mucho en común cuando son escritos a mano.

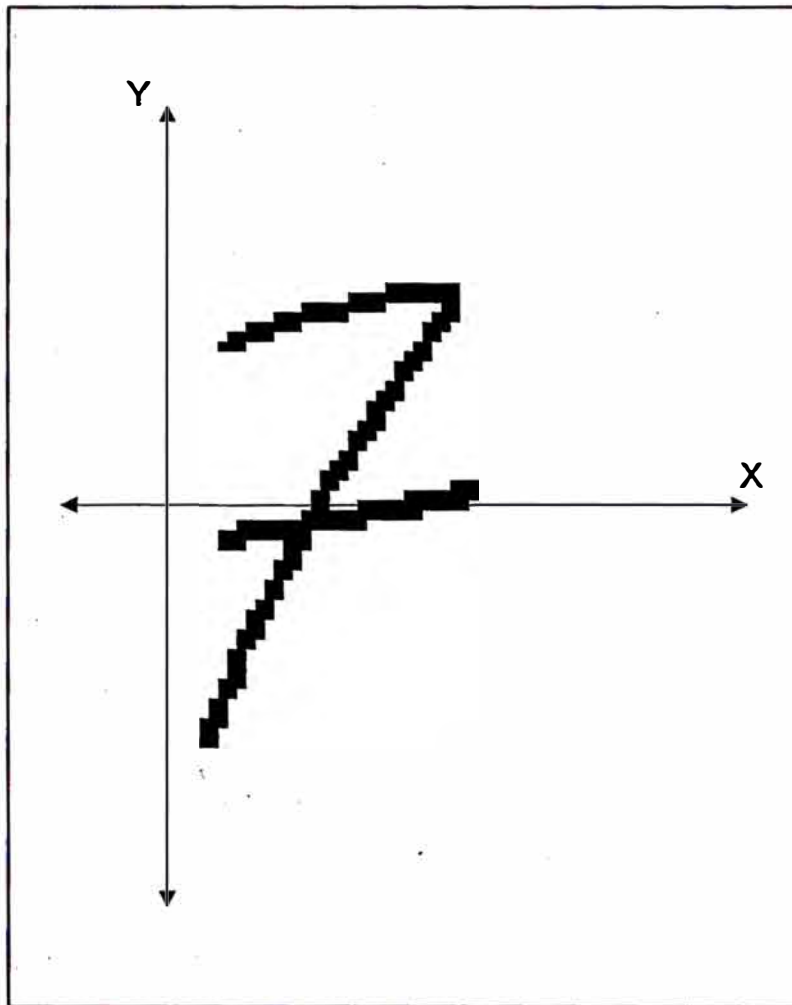
Por propósitos prácticos hemos determinado el uso de solamente dos características por muestra, esto nos permitirá la visualización en un plano cartesiano la ubicación de las distintas muestras, dicha determinación no restó generalidad a la solución.

El plano cartesiano es importante pues nos permite observar gráficamente la ubicación de cada uno de los pixels que conforman el dígito. La imagen aquí presentada se ha simplificado para obtener una mejor visualización de la idea; el algoritmo que se desarrolló para obtener el vector de características toma una matrix de 64 x 64.

La gráfica resultante para cada alternativa nos permitirá efectuar la selección. Lo que buscamos es una separación lineal de los espacios formados por el conjunto de muestras de los dígitos 1 y el conjunto de muestras de los dígitos 7

### **4.3.3 Alternativas**

Para una mejor definición de las características es necesario tener presente el trazado del sistema cartesiano mostrado en la Figura 4.2 en el cual se coloca a la matriz de representación del dígito en los cuadrantes “I” y “IV” del mismo.



**Figura 4.2 Plano para obtener las características**

Cada cuadrado representa un pixel donde:

‘.’: Pixel Negro; pixel pertenece al dígito

‘ ‘: Pixel Blanco; pixel no pertenece al dígito

**Primera Alternativa:**

Definición de la Característica x1:

x1: Abcisa del centroide de los Pixels Negros.

Definición de la Característica x2:

x2: Ordenada del centroide de los Pixels Negros.

**Segunda Alternativa:**

Definición de la Característica x1:

x1: ratio que mide la distancia promedio de todos los Pixels Negros ubicados encima del eje X (ordenada positiva) entre la distancia promedio de todos los Pixels Negros ubicados por debajo del eje X (ordenada negativa).

Definición de la Característica x2:

x2: ratio que mide la distancia promedio desde el punto fijo localizado en (0,0) a los Pixels Negros que están por encima del eje x entre la distancia promedio desde el mismo punto a los Pixels Negros que están por debajo de dicho eje.

**Tercera Alternativa:**

Definición de la Característica x1:

x1: Porcentaje de Pixels Negros ubicados encima del eje X (ordenada positiva) del total de Pixels encima de dicho eje.

Definición de la Característica x2:

x2: Porcentaje de Pixels Negros ubicados por debajo del eje X (ordenada negativa) del total de Pixels encima de dicho eje.

#### **4.3.4 Elección Realizada**

Hemos elegido la segunda alternativa por ser la que mejor cumple con los requisitos establecidos inicialmente.

### **4.4 Elección del vector de características para entrenar al SOFM**

#### **4.4.1 Objetivo**

Obtener el vector de características que mejor describa a cada uno de los dígitos.

#### **4.4.2 Condiciones a cumplir**

Tenemos que obtener 10 vectores de características patrón, una para cada dígito con la condición de que sean similares para cada conjunto de muestras de un mismo dígito pero diferentes de los vectores de características de los otros dígitos tomados como muestras.

El vector de características representan la información que la Red Neuronal debe “aprender”; se debe obtener dicho vector para cada uno de los dígitos que se utilizarán en el entrenamiento de la red.

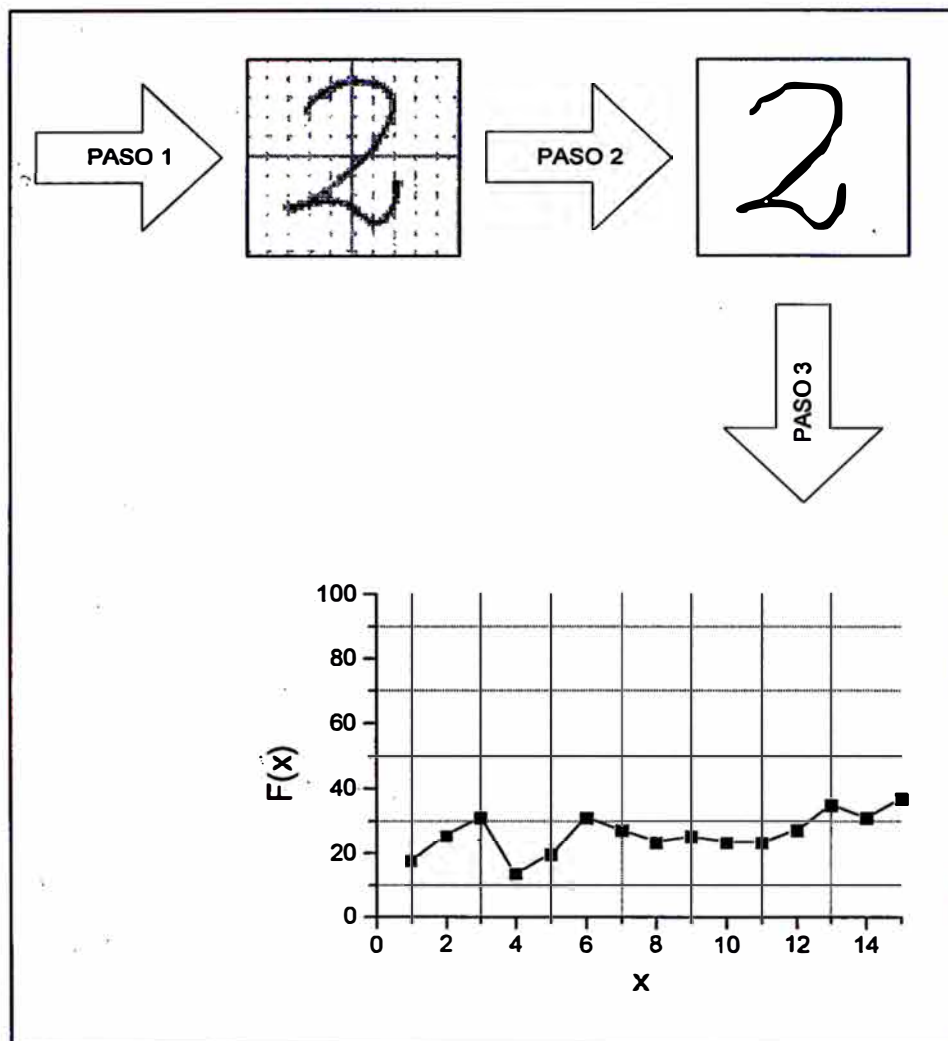
La extracción del vector de características (feature vector) constituye la pieza fundamental para la implementación de nuestra solución basada en el Mapa Auto-Organizativo de Kohonen.

El éxito de nuestra implementación descansa en una buena elección de dicho vector.

Nuestra interfase nos permite obtener el vector de características en un formato entendible por la red neuronal, es decir, transformar la imagen digitalizada del dígito en un vector con sus características.

#### **4.4.3 Alternativas para la extracción del Vector de Características (feature vector)**

En la figura 4.3 mostramos el esquema integral del preprocesamiento llevado a cabo para la obtención del vector de características a utilizar.



**Figura 4.3 PREPROP**

Hemos denominado PREPROP a dicho procedimiento, el cual consta de tres partes. Los dos primeros pasos describen gráficamente el algoritmo del punto 4.2 por lo que los mencionaremos muy brevemente:

**Primer Paso:**

Obtener la imagen del dígito en formato BMP de 16 tonos de gris.

**Segundo Paso:**

Obtener la imagen del dígito en formato BMP monocromático.

### Tercer Paso:

En este paso se obtiene el vector de características del dígito; teniendo como entrada la imagen obtenida en el paso 2. obtenemos una representación matricial de la misma.

Regla casi general al momento de escribir los dígitos: la escritura de los dígitos es de una manera continua (no ocurre en el siete, cuatro y a veces en el uno y el cinco), es decir, con un solo trazo; esto quiere decir que si por ejemplo separamos al dígito en partes mediante cortes verticales, nos daremos cuenta que cada parte guarda una estrecha relación con la parte que está a su izquierda y también a su derecha.

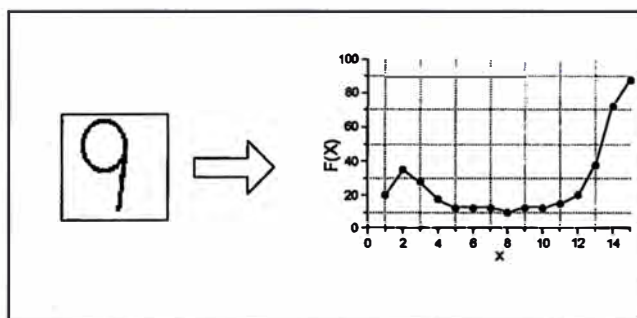
Tomar columnas de pixels y calcular el porcentaje de Pixels Negros dentro del área considerada, esto nos permite uniformizar los valores (no interesa el ancho del trazo del dígito pues guarda armonía con el tamaño del mismo); para la siguiente medida consideramos algunas columnas de pixels de la lectura anterior y añadimos nuevas columnas de pixels, formando una nueva área y procedemos a tomar una siguiente medida.

Para una mejor obtención de valores hacer un corrimiento de los dígitos, es decir, empezar los cálculos a partir de la columna donde se encuentren los primeros Pixels Negros.

Como no todos los dígitos tienen el mismo ancho, el conjunto de columnas de pixels tiene que ser variable a fin de uniformizar el número de lecturas, aquí consideramos columnas de tres, cuatro y hasta cinco pixels; los retrasos (conjunto de columnas de pixels de la lectura anterior) están en función al número de pixels elegidos y fue de alrededor del 30%.



De esta forma obtuvimos 15 lecturas para cada uno de los dígitos. En la Figura 4.4 podemos apreciar las 15 lecturas obtenidas para el dígito nueve (9)



**Fig. 4.4 Gráfica del Vector Característico del 9**

## **Capítulo 5: Estrategia de Solución para el entrenamiento de los modelos**

### **5.1 Alcances**

Tal como se menciona al principio tenemos que hacer una adecuación a la implementación real. Nuestro modelo de red neuronal es una implementación algorítmica que simulará las características que debe presentar dicha red.

El principal problema queda resuelto al tener como entrada datos simbólicos lo que permite una solución algorítmica basada en computadoras digitales; existe en el mercado software que permite este tipo de implementaciones tal como el Neural Networks Toolbox de MatLab, pero hemos preferido desarrollar uno propio a fin de controlar la integración del sistema global; hemos utilizado dicho software para la validación de nuestro modelo.

Las características de la red neuronal son descritas en cada una de las implementaciones llevadas a cabo. Los datos simbólicos que ingresan al modelo son denominados vectores característicos, es decir, la información representativa de cada uno de los dígitos que ingresan al modelo durante el proceso de entrenamiento y también durante el proceso de reconocimiento. Por motivos puramente prácticos se decidió que la salida sea sólo simbólica para así lograr una integración más efectiva con sistemas que puedan utilizar el resultado obtenido luego del reconocimiento efectuado por la red neuronal.

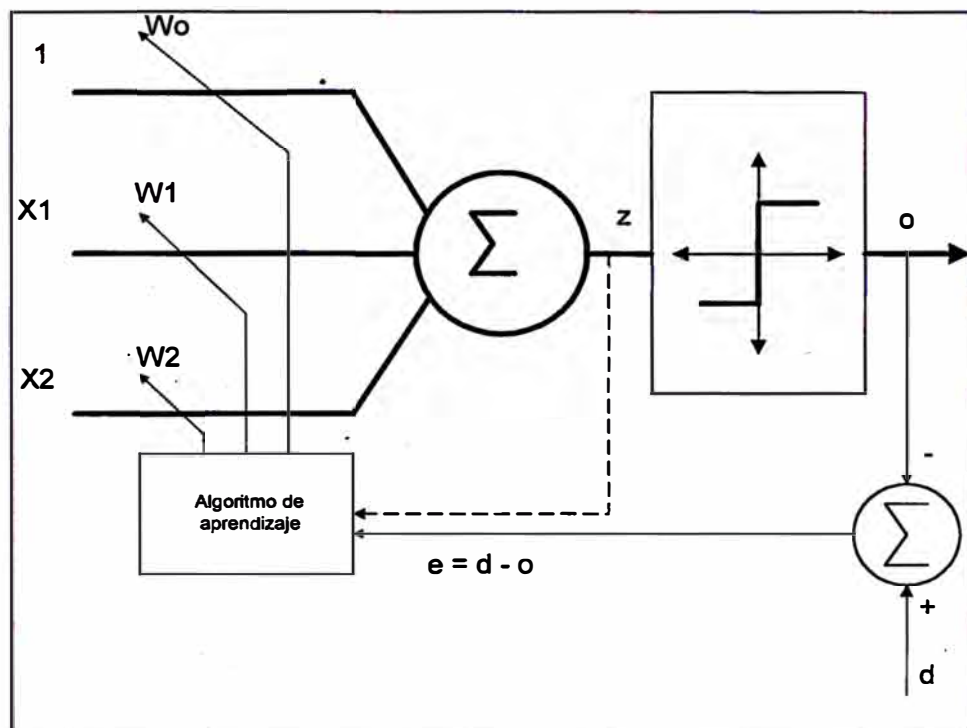
### **5.2 Una aproximación inicial haciendo uso del Perceptron**

El modelo nos servirá para entrenar una red neuronal que sea capaz de reconocer los dígitos uno (1) y siete (7).

### 5.2.1 El modelo

Hemos elegido para esta primera implementación el modelo de red neuronal conocido como PERCEPTRON.

La Figura 5.1 nos muestra el modelo de red neuronal utilizado; cabe mencionar que sólo se usa una neurona para esta implementación.



**Figura 5.1 Implementación mediante el uso del Perceptron**

Donde:

<b>1</b>	Constante que indica que el umbral $w_0$ no sufrirá variación durante el entrenamiento.
<b>x1</b>	Valor de la primera característica del dígito elegido
<b>x2</b>	Valor de la segunda característica del dígito elegido
<b>w0</b>	Valor del Umbral (también es un peso)
<b>w1</b>	Peso para la conexión con la primera característica
<b>w2</b>	Peso para la conexión con la segunda característica
<b>z</b>	Suma ponderada de las entradas y sus pesos
<b>Función Signo</b>	Convierte el valor $z$ a $o$
<b>o</b>	Valor Obtenido $o = 1$ : El modelo asume que la entrada corresponde al 1 $o = -1$ : El modelo asume que la entrada corresponde al 7
<b>d</b>	Valor Deseado $d = 1$ : El valor del dígito elegido es 1 $d = -1$ : El valor del dígito elegido es 7
<b>e</b>	Coefficiente de error calculado; diferencia entre Valor Deseado y Obtenido

Como se aprecia, el modelo planteado consta de tres entradas, la primera que siempre tiene como valor la constante "1" y las otras dos son los valores de las características  $x_1$  y  $x_2$  de cada uno de los dígitos que ingresan al entrenamiento.

Características de la red neuronal:

Número de Neuronas	<b>1</b>
Número de Características	<b>2</b>
Número de Pesos	<b>3</b>

Parámetros para el aprendizaje:

Coefficiente de aprendizaje ( <b>etha</b> )	<b>0.04</b>
Tiempo de aprendizaje	<b>200</b>
Regla de Aprendizaje	<b>Widrow-Hoff</b>
Error a considerar	<b>0.001</b>

Con estos valores logramos la estabilización del modelo.

Los valores de los pesos antes del entrenamiento:

w0	0.73280
w1	0.68400
w2	0.19620

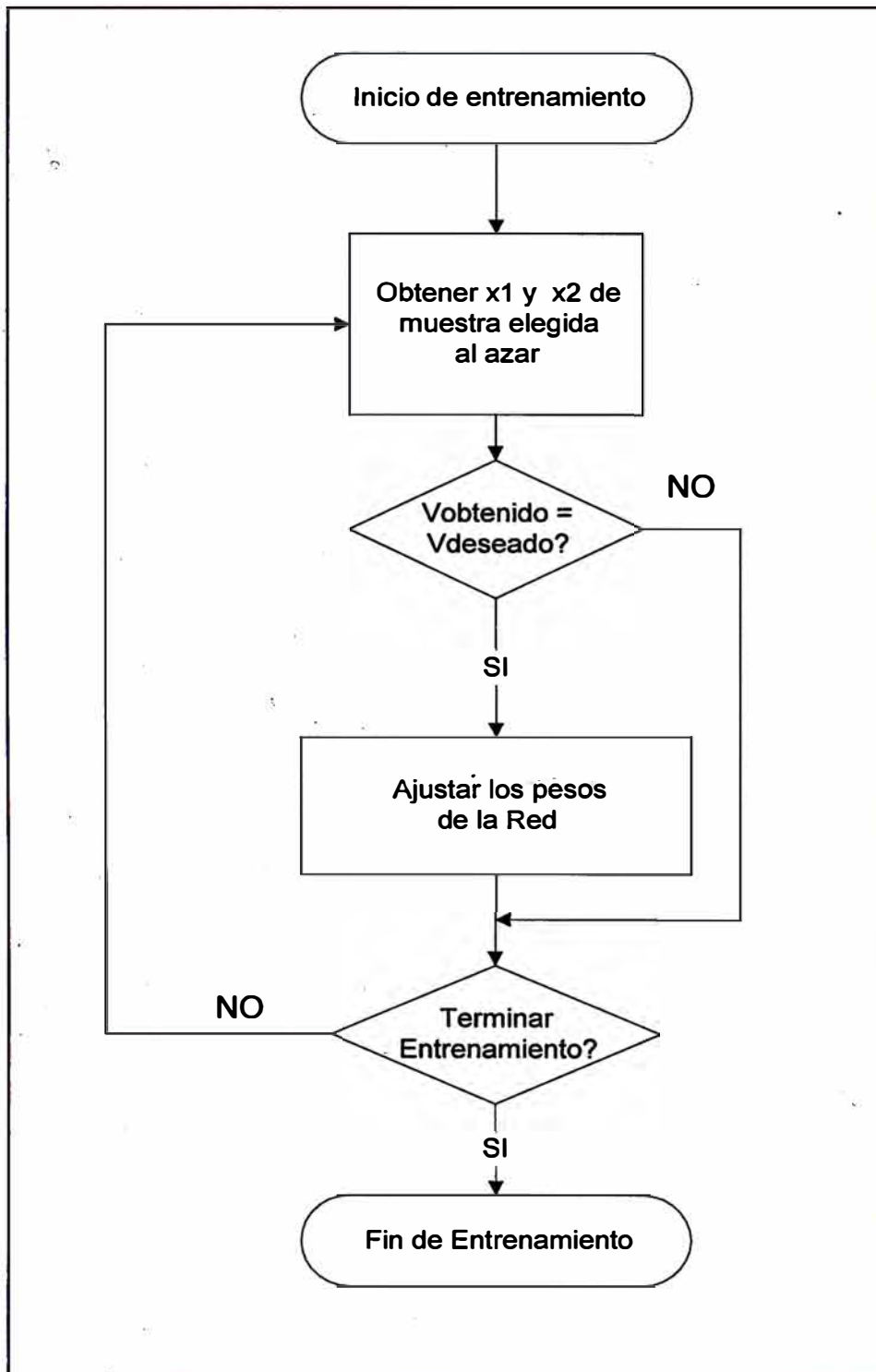
**Muestras**

En el anexo 1 mostramos la relación de valores  $x_1$ ,  $x_2$  de los dígitos 1 y 7 utilizados en el entrenamiento, así como los valores deseados para cada uno de ellos.

**Algoritmo de entrenamiento:**

- 1 Fijar parámetros de entrenamiento
- 2 Inicializar los pesos
- 3 Obtener entradas  $x_1$ ,  $x_2$  y Valor Deseado (1 ó 7)
- 4 Calcular Suma Ponderada y Valor Obtenido
- 5 Actualizar pesos en función al Valor Obtenido y Valor Deseado
- 6 Si no se ha completado el entrenamiento regresar al paso 3.

En la Figura 5.3 mostramos el diagrama de bloques para llevar a cabo el proceso de entrenamiento de la red. .



**Figura 5.3 Diagrama de bloques del entrenamiento**

Explicación de los pasos:

## **1 Fijar parámetros de entrenamiento**

Como se explicó anteriormente no existen reglas definidas para la determinación de estos parámetros, las consideraciones son experimentales. Los valores presentados aquí (tiempo y constante de aprendizaje), fueron obtenidos luego de una serie de corridas efectuadas al modelo, y son las que a nuestro juicio nos permiten obtener los pesos finales adecuado para tener buenos resultados al momento de usar la red para el reconocimiento de los dígitos, luego de haber logrado la estabilización del modelo.

Definidos anteriormente. Según se menciona no hay un procedimiento definido para la determinación de dichos parámetros; nos atrevemos a mencionar que la regla es de “ensayo y error”, lo cual nos hace suponer la cantidad de pruebas a realizar antes de obtener resultados satisfactorios.

## **2 Inicializar los pesos**

De acuerdo a las características del modelo asignamos valores aleatorios a los pesos  $w_0$ ,  $w_1$  y  $w_2$  con lo cual estamos asegurando que el conocimiento inicial del modelo es “cero” con respecto a la información que debe almacenar para llevar a cabo su tarea.

### **Obtener entradas $x_1$ , $x_2$ y Valor Deseado (1 ó 7)**

Seleccionamos aleatoriamente un par ordenado  $(x_1, x_2)$  del conjunto de muestras utilizado así como el Valor Deseado.

### **Calcular el Suma Ponderada y Valor Obtenido,**



$$z = w1*x1 + w2*x1 - w0;$$

o = Función Rampa(z);

## **5 Actualizar pesos en función al Valor Obtenido y Valor Deseado**

Dicho ajuste se realizó siguiendo la regla de aprendizaje propuesto por Widrow y Hoff; de acuerdo a esta regla se obtiene el valor producto de los cálculos con los pesos actuales y se compara contra el valor deseado (el valor que debería ser en función al dígito que en ese momento se está utilizando), si es que existe mucha diferencia significa que la red todavía no se ha estabilizado por lo que hay que elegir al azar un nuevo dígito y volver al primer módulo; de existir poca diferencia nos atrevemos a decir que la red se ha estabilizado, dando por terminado la fase de entrenamiento de la red neuronal.

Es necesario mencionar que siempre existirá una diferencia entre el valor obtenido y el valor deseado, por lo que es necesario manejar un error tal y como lo plantea la regla de aprendizaje. Para nuestros propósitos hemos considerado un error del orden de un milésimo ( $10^{-3}$ ).

$$w1 = w1 + \text{etha}*(d-o)*x1;$$

$$w2 = w2 + \text{etha}*(d-o)*x2;$$

## **6 Si no se ha completado el entrenamiento regresar al paso 3**

El entrenamiento de la red es un proceso iterativo determinado por el tiempo de aprendizaje establecido, donde en cada unidad de tiempo se presenta de manera aleatoria las características de la muestra presentada y la actualización de los pesos está en función a la constante de aprendizaje determinada previamente.

### 5.2.2 Resultados Obtenidos

Los valores de los pesos al final de entrenamiento:

w0	0.73280
w1	0.55050
w2	0.13016

Ecuación de la recta de decisión:

$$y = mx + b$$

donde:

$m = -w1/w2$	-4.22941
$b = w1/w2$	5.63000

$$m = -w1/w2 = -4.22941$$

$$b = w1/w2 = 5.63000$$

Dicha ecuación es la frontera que divide el espacio asignado a los dígitos 1 y el espacio asignado a los dígitos 7. Con esto hemos concluido el proceso de aprendizaje.

En el anexo 3 se muestran los valores finales de los pesos luego de llevado a cabo el entrenamiento.

### 5.2.3 Conclusiones

De la gráfica y de los valores obtenidos podemos concluir que el modelo a logrado estabilizarse y se encuentra listo para llevar a cabo la tarea de reconocimiento y posterior clasificación.

Asímismo de la gráfica concluimos que el modelo presenta cierta distorsión pues encontramos dígitos 1 en la zona de los dígitos 7 y viceversa; dicha anomalía se presenta bastante cerca a la frontera de decisión. Esto nos permite afirmar que en la práctica la frontera de decisión no es una línea recta sino una banda, es decir, existe una zona donde la incertidumbre es máxima. Para nuestro caso la banda es pequeña por ende la eficiencia del modelo es aceptable. La presencia de esta banda o indicativo de zona de incertidumbre nos explica el hecho de que muchos dígitos 1 o 7 son fácilmente confundidos aún por nosotros mismos, característica encontrada en este tipo de modelos.

#### **5.2.4 Limitaciones del Modelo**

La principal limitación nos la da el mismo modelo, pues como mencionábamos es necesario tener información a priori acerca de si es posible o no trazar una frontera lineal de decisión, de no ser así no se puede lograr una estabilización durante el proceso de entrenamiento y por ende no se pueden obtener los valores finales de los pesos que nos aseguren que el modelo ya está listo para ser utilizado. Existen técnicas alternativas para salvar el problema de la frontera lineal de decisión, pero no es el tema que nos compete enfocar.

Otra limitante es que el modelo no tiene capacidad de detectar si es que el dígito a reconocer es un uno (1) o un siete (7) pues solamente actúa como un bloque IF-THEN-ELSE , esto se debe a que divide todo el espacio en dos grandes regiones asumiendo que solamente existen dígitos uno y siete, por lo que podrían obtenerse resultados no deseados; una forma de salvar este problema es mediante un “filtrado” previo de la información lo cual hay que tener presente al momento de efectuar una implementación real.

Un esquema general de solución es descrito en: [2] “Handwritten Digit Recognition by Neural Networks with Single-Layer Training”; donde utilizan el modelo de red neuronal conocido como Perceptron Multicapa (Multilayer Perceptron MLP) que son varios Perceptrones dispuestos en diferentes capas, adicionalmente se hace uso del algoritmo de Retropropagación (backpropagation algorithm) para el entrenamiento de la red. Este algoritmo es el más popular para el entrenamiento supervisado de las redes neuronales artificiales y fue propuesto por Paul Werbos.

### **5.3 El modelo mediante el SOFM**

#### **5.3.1 Antecedentes**

Las máquinas lectoras de cheques nos proporcionan la solución inicial al problema planteado, sin embargo mantienen un alto componente de trabajo manual para la captura de datos dentro de los cheques a fin de efectuar los procesos de conciliación respectivos.

El proceso manual incrementa los costos de procesamiento e inclusive imagen de la institución por los múltiples problemas en la forma de reclamos a los cuales hacer frente.

Un componente que viene a complicar la operatividad fue el incremento en el uso de los cheques, dada su popularidad como instrumento de negocio. Un número mayor de transacciones tuvieron que ser procesadas lo cual significó el incremento sustancial en los costos de procesamiento.

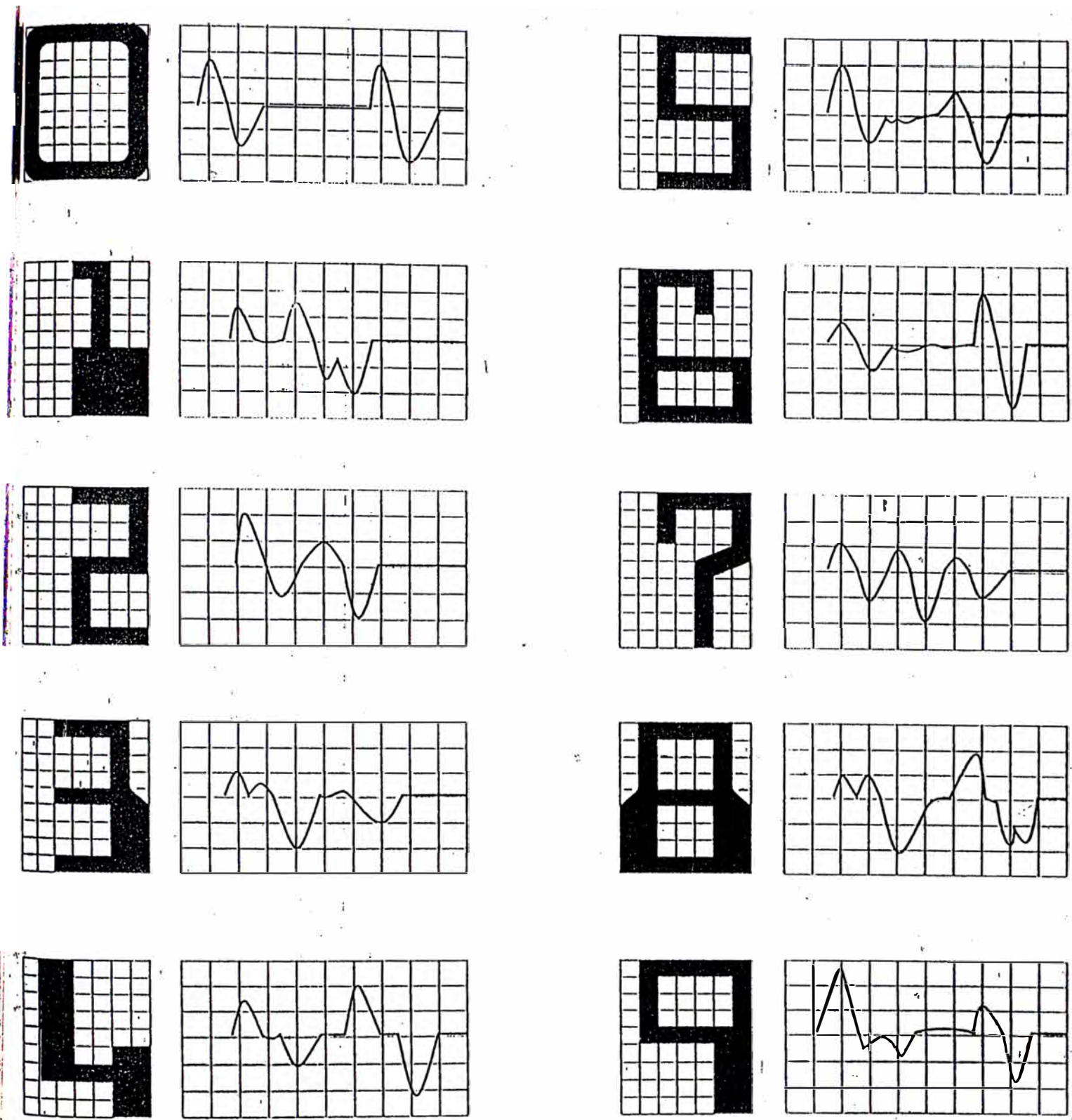
Por ende el problema se convierte en una necesidad para las empresas sobre todo financieras y constantemente se hicieron esfuerzos por resolverlos.

Un paso decisivo de aproximación a la solución lo encontramos en la iniciativa que tuvieron los Bancos Norteamericanos al reunirse para resolver conjuntamente los problemas que afectaban a todos. El principal acuerdo fue la

estandarización del tamaño y ubicación de la información a consignar en el cheque a fin de permitir una mejor manipulación del mismo inicialmente manual y posteriormente automático. Una de estas modificaciones comprende la incorporación de espacios asignados para identificar el número del cheque, la cuenta asociada al cheque, el banco al que pertenece el cheque. Como mencionamos el siguiente paso consistía en la lectura automática de dicha información.

Resolver este problema implica un reconocimiento de los dígitos y caracteres especiales (a los cuales denominaremos **tipos** de aquí en adelante); dada la gran variedad de formas de los dígitos se convino en efectuar una estandarización de los mismos, es decir, la creación de un patrón único que permita automatizar la lectura.

Se utilizó una matriz de tamaño fijo (7 columnas por 9 filas) para la representación de los tipos. Para la impresión de los tipos se utilizó tinta magnética y se tuvo que diseñar el correspondiente lector magnético. Dicha lectora realiza la lectura columna a columna y detecta la variación de área magnetizada entre la lectura actual y la lectura anterior. En la Figura 5.4 se muestra la gráfica de los tipos y sus correspondientes lecturas.



**Fig. 5.4 Conjunto de Tipos y sus curvas características**

Los tipos patrón corresponden a dígitos deformados de tal forma que las curvas características sean lo bastante diferentes unos de otros; esto permite mejores resultados en el proceso de reconocimiento automático.

El proceso de reconocimiento consta de dos etapas:

- 1) Cálculo de la curva característica del tipo “desconocido” .
- 2) Matching entre la curva característica obtenida y las curvas características almacenadas previamente, a fin de determinar a cual de ellas corresponde y por ende el tipo asociado con la misma.

Las máquinas lectoras de cheques siguen este principio.

El problema de la lectura de los importes era mayor, sobre todo para los cheques girados por personas naturales dado que eran escritos en forma manual; dada las limitaciones, se quedó como un problema sin resolver. Se pudo salvar parte del problema mediante la preparación de cheques con el importe escrito con los mismos tipos y ubicados en una zona específica dentro del cheque; esto permitía su lectura pero no resuelve el problema de manera integral.

### **5.3.2 El modelo**

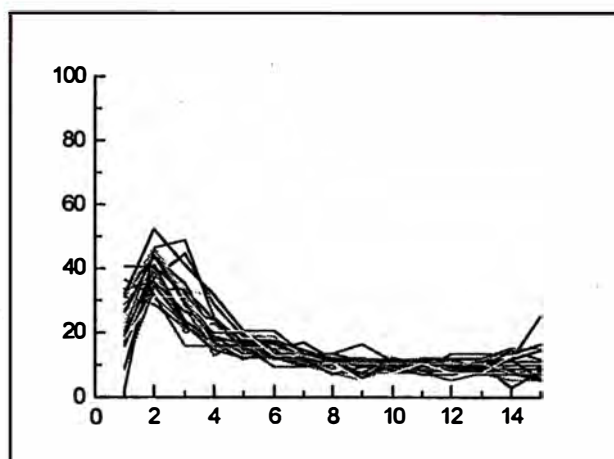
Se enfoca la solución mediante una combinación de la técnica descrita anteriormente y el empleo del modelo de red neuronal artificial conocido como el modelo de Kohonen o Mapa Auto-Organizativo (SOFM: Self-Organizing Feature Map)

#### **Vectores de Características utilizados en el Entrenamiento**

Se han utilizado 20 muestras de cada dígito; cada curva característica es representada por un color; se ha preferido colocar todas las curvas características en un solo gráfico a fin de que se pueda apreciar gráficamente la similitud existente.

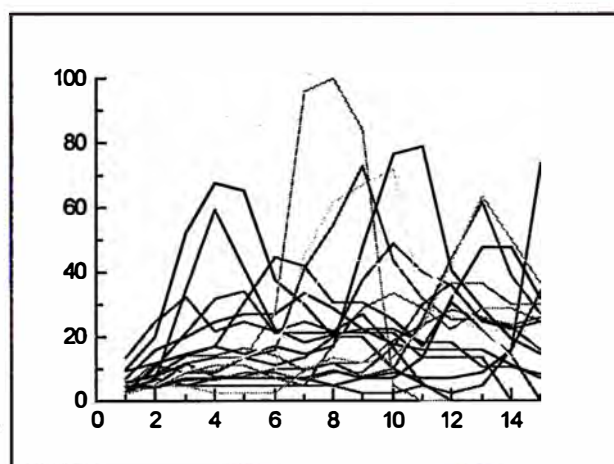
En el eje de las ordenadas se grafica el porcentaje de área perteneciente al dígito según la forma de lectura definida previamente. Se han efectuado 15 lecturas para cada uno de los dígitos.





**Fig. 5.5.0 Curvas Características del '0'**

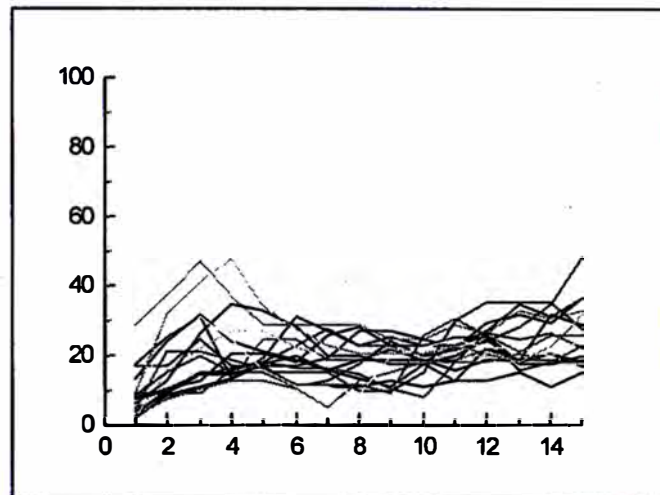
En este gráfico podemos apreciar la gran similitud existente entre todas las curvas características, esto se debe a que no existe mayor diferencia en la escritura de dicho dígito, es decir, casi todos escribimos este dígito de una manera muy parecida. Las curvas características son independientes del tamaño del tamaño de los dígitos.



**Fig. 5.5.1 Curvas Características del '1'**

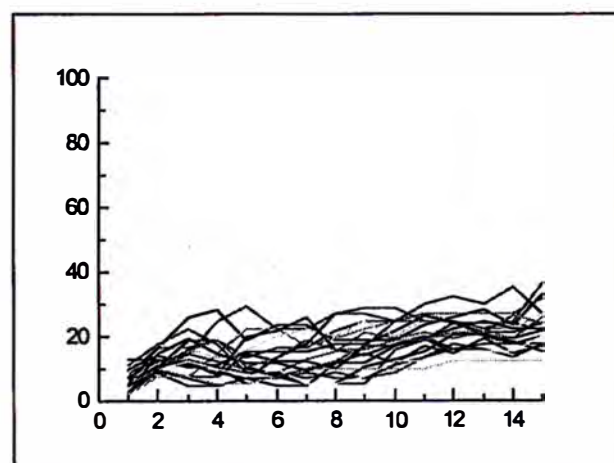
En este gráfico no se aprecia a simple vista la similitud entre las diferentes curvas características, sin embargo si se observa detenidamente nos daremos

cuenta que la similitud se da por grupos. Al principio pensamos que se estaba cometiendo un error; luego del proceso de aprendizaje descartamos dicha posibilidad por los resultados obtenidos.



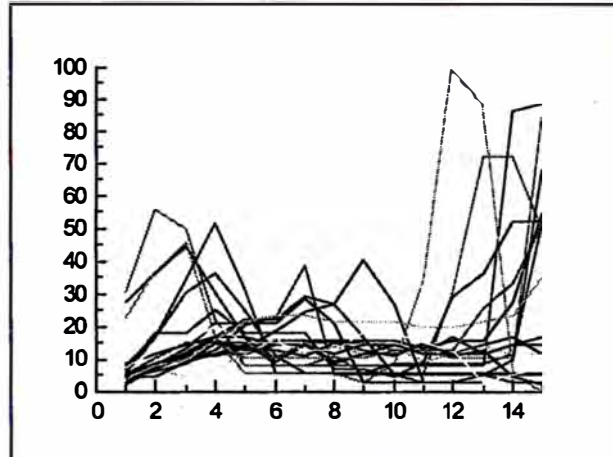
**Fig. 5.5.2 Curvas Características del '2'**

Lo mismo sucede en este gráfico, la forma de cálculo de las curvas características absorbe muy bien las diferencias en la forma de escritura de los dígitos '2'.



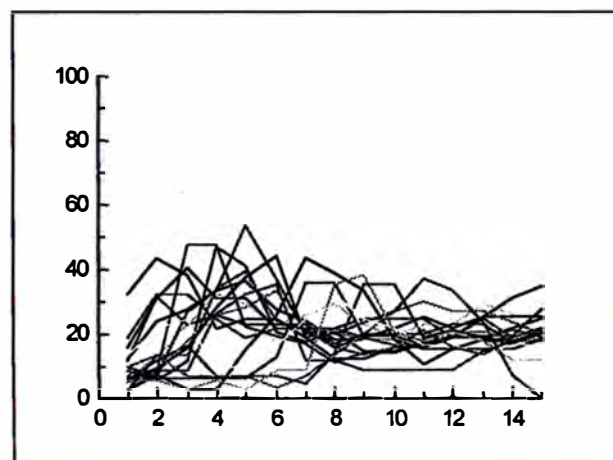
**Fig. 5.5.3 Curvas Características del '3'**

En igual forma, las diferencias entre los dígitos '3' son absorbidas por la forma de cálculo de las curvas características.



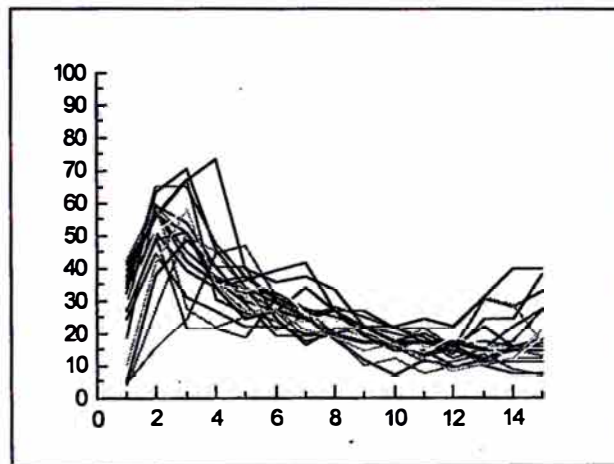
**Fig. 5.5.4 Curvas Características del '4'**

Al igual que en los dígitos '2' se observan similitudes por grupos, hecho que es asimilado eficientemente por el modelo propuesto. Puede notarse en la parte final de las curvas que el porcentaje del área perteneciente al dígito bordea el 100%, esto obedece a la forma que tiene el dígito.



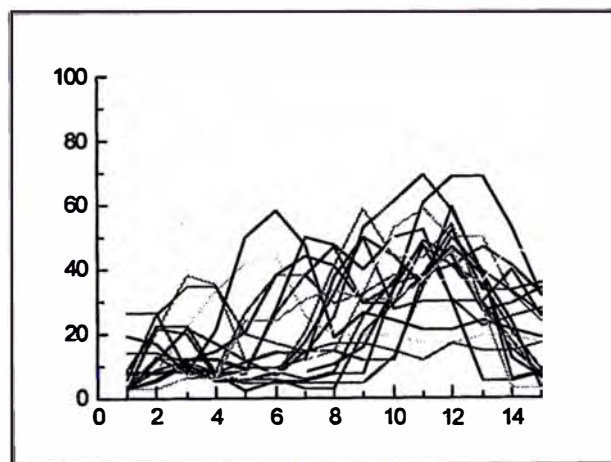
**Fig. 5.5.5 Curvas Características del '5'**

Al igual que los dígitos '4' encontramos similitudes por grupos. Podemos apreciar que en promedio el área perteneciente a los dígitos bordea el 50%, esto nos indica que dicha área es casi constante, hecho que es explicado por la forma del dígito.



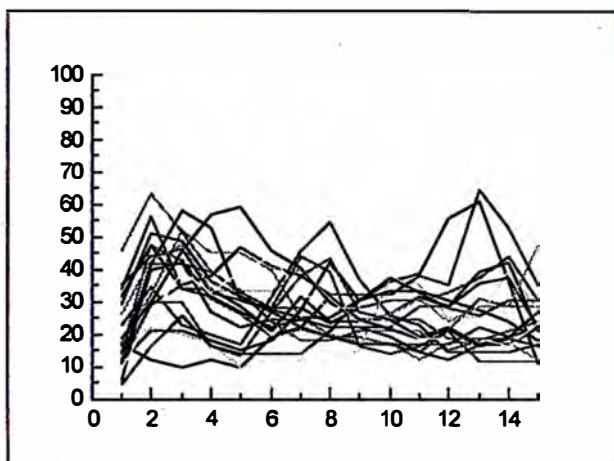
**Fig. 5.5.6 Curvas Características del '6'**

La similitud entre los dígitos es bastante alta.



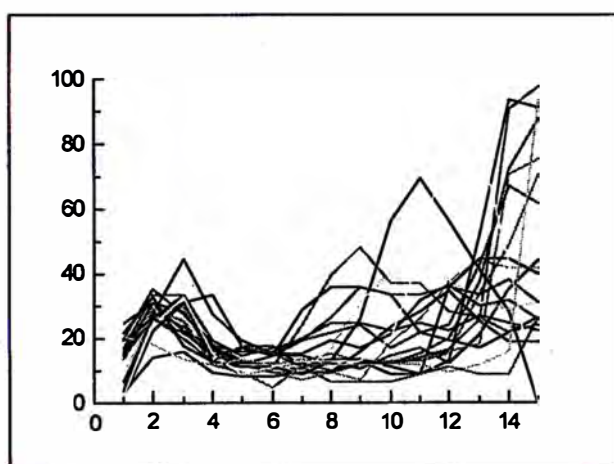
**Fig. 5.5.7 Curvas Características del '7'**

La similitud entre los dígitos se presenta por grupos, esto es explicado por la diversidad de formas al momento de escribir el dígito '7' especialmente en el uso o no de la raya del medio.



**Fig. 5.5.8 Curvas Características del '8'**

Al igual que en las gráficas del dígito '5' vemos una constante a lo largo del gráfico, explicado por la forma del dígito.



**Fig. 5.5.9 Curvas Características del '9'**

Es interesante notar aquí que los resultados del cálculo de las curvas características son el reflejo de las curvas características de los dígitos '6' si es que tomamos como eje de las ordenadas como el eje de reflexión.

### **Parámetros del modelo elegido**

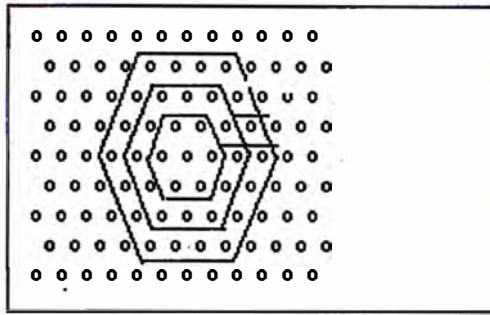
La principal dificultad en la implementación de soluciones mediante el empleo de redes neuronales artificiales radica en la falta de reglas precisas en cuanto a la elección de los vectores de características (feature vector), el tamaño de la red, el número de conexiones y los parámetros de entrenamiento.

#### Características de la red:

Forma de la Red	<b>Hexagonal de 2 capas</b>
Primera Capa	<b>15 entradas</b>
Segunda Capa	<b>400 neuronas</b>
Cantidad de Conexiones	<b>6,000</b>
Cantidad de Pesos	<b>6,000</b>

#### Forma de la Red:

En empleo de la red hexagonal nos permite mayor asociación entre la información que almacena una neurona y sus neuronas vecinas. La Figura 5.6 nos muestra un ejemplo de la forma de la red neuronal a utilizar.



**Figura 5.6 Red Hexagonal**

Primera Capa:

Las quince entradas son utilizadas para recibir la información de las quince lecturas obtenidas para cada uno de los dígitos.

Segunda Capa:

Para nuestros propósitos hemos utilizado 400 neuronas que almacenarán la información de las 200 muestras (20 por cada dígito) utilizados en el proceso de entrenamiento. A esta capa también se le denomina Mapa Topográfico bidimensional y para nuestro modelo hemos dispuesto que sea hexagonal de 20 filas por 20 columnas

Cantidad de Conexiones:

Determinado por la cantidad de neuronas en la primera y segunda capa.

Cantidad de Pesos:

Determinado por la cantidad de conexiones de la red elegida.

**Parámetros para el entrenamiento**

Los parámetros para el entrenamiento fueron elegidos de acuerdo a las reglas prácticas descritas por Kohonen:

Elección de la muestra:

Elegida aleatoriamente de las 200 muestras disponibles a lo largo de todo el proceso de entrenamiento.

Tiempo de entrenamiento:

Se asume que el tiempo de entrenamiento es una función del número de neuronas de la red (segunda capa), para nuestro modelo hemos utilizado hasta 30,000 iteraciones. Una de las características que se le incorporó a nuestro programa fue la de permitir observar gráficamente el aprendizaje que se está llevando a cabo; de esta forma podemos ver, cada cierto número de iteraciones, el arreglo de los dígitos en el mapa topográfico. Adicionalmente el programa imprime el valor de los pesos de las conexiones cada cierto número de iteraciones lo que nos permite evaluar más rápidamente los resultados obtenidos.

Coefficiente de Aprendizaje:

Este coeficiente varía según el número de iteración en el cual nos encontremos durante el proceso de entrenamiento. En las primeras 1,000 iteraciones decrece linealmente de 1 a 0.9; para las siguientes 9,000 iteraciones decrece de 0.9 a 0.5 y para las siguientes 90,000 iteraciones decrece de 0.5 a 0.001. Como se mencionó anteriormente este coeficiente es una medida de en cuanto hay que actualizar el conocimiento previamente adquirido.



### Radio de la vecindad en el tiempo de entrenamiento:

En la Figura 5.6 también podemos apreciar la forma como se hace la asignación de este parámetro durante el proceso de entrenamiento; al igual que el parámetro anterior el eje de las ordenas nos indica el valor que toma para cada tiempo  $t$ . En el entrenamiento empezamos con un radio igual a 6, con un decrecimiento de una unidad cada 5,000 iteraciones durante las primeras 25,000 iteraciones; posteriormente el radio de la vecindad se mantiene en 1 hasta terminar con el entrenamiento.

### **Entrenamiento**

Existen muchos paquetes comerciales y de libre distribución para llevar a cabo el entrenamiento de redes neuronales artificiales. A fin de tener un mayor entendimiento acerca de la forma como es que funciona el algoritmo, hemos codificado el mismo haciendo uso del lenguaje C; suponemos que es una contribución producto de este trabajo y adicionalmente logramos con ellos una mayor flexibilidad como componente fácilmente integrable a una solución comercial.

Parte de la validación del programa que codificamos fue realizado por el grupo del Neural Networks Research Center (NNRC) del Helsinki University of Technology en Finlandia, los contactos fueron efectuados por medio de Internet lo cual nos permitió corregir algunos errores conceptuales en muy poco tiempo, que de otra forma habría tomado mucho tiempo utilizando los medios de comunicación convencionales.

Las corridas de prueba del modelo fueron realizadas en un supercomputador en la misma Universidad. Cabe mencionar que los parámetros arriba mencionados obedecen a sugerencias efectuadas por personas que están trabajando el modelo para otras aplicaciones lo cual consideramos es un buen punto de partida.

Adicionalmente realizamos numerosas pruebas en diferentes tipos de Personal Computers (PC 's) compatibles.

### **Tiempo que demora el Entrenamiento**

Como datos estadísticos cabe mencionar lo siguiente:

Tiempo de Aprendizaje	<b>30,000 iteraciones</b>
PC-386 DX 40 Mhz	<b>6 horas</b>
PC-486 DX 100 Mhz	<b>15 minutos</b>
PC-Pentium 166 Mhz	<b>10 minutos</b>
Cray	<b>45 segundos</b>

El tiempo que demora el entrenamiento está determinado por la cantidad de iteraciones que hay que llevar a cabo, el tamaño de la red neuronal propuesta (número de neuronas y su disposición), el algoritmo utilizado y el ciclo del reloj de la CPU de la máquina donde se efectuará. El tamaño de la red fue elegido de antemano por lo que podemos considerarlo como constante a fin de medir el tiempo de demora en el entrenamiento.

Es posible lograr una buena implementación mediante el uso de técnicas adecuadas de programación; los cálculos básicos llevados a cabo son: cálculo del cuadrado de la distancia y matching para determinar el cuadrado del error mínimo;

Todos los cálculos son efectuados en punto flotante por lo que es necesario utilizar un lenguaje que lo maneje de manera eficientes.

Queda por resolver la unidad de almacenamiento a utilizar para los números: vectores o punteros, se eligió el uso de vectores por permitir una implementación no tan eficiente pero si sencilla de programar ; no se hizo el

cálculo en la mejora en el tiempo de aprendizaje si se utilizaba punteros pues no es el motivo de la tesis, sin embargo, las modificaciones al programa quedan para mejoras al modelo en la parte del aprendizaje de la red.

Otro punto por resolver está determinado por el ciclo del reloj de la CPU, a mayor potencia, menor tiempo de aprendizaje. Este tipo de implementaciones requieren el uso de máquinas que procesen la mayor cantidad de instrucciones por unidad de tiempo, requisito indispensable para futuras aplicaciones que se quiera desarrollar. Al no contar con máquinas de “buena capacidad” se optó por utilizar varias PC’s cada una efectuando el entrenamiento de manera independiente; de esta forma se obtenían varios resultados en menor tiempo.

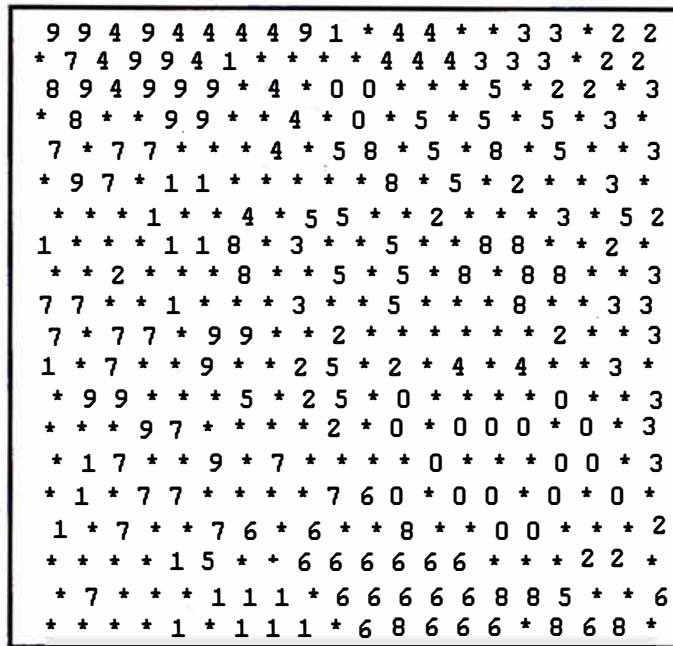
### **Estabilización del modelo**

A fin de ir midiendo la estabilización del modelo efectuamos diversas corridas al programa con diferentes números de iteraciones; logramos una estabilización aceptable para 30,000 iteraciones.

Dicha estabilización está basada en el hecho de que una vez entrenada la red y con los pesos así establecidos es utilizada para reconocer los 200 dígitos que se utilizaron como muestra. Una medida de eficiencia es obtenida mediante el promedio de aciertos del total de pruebas que para nuestro caso fue de 95%.

### **Valores finales de los pesos**

En el Anexo 5 se muestran los valores finales de los pesos luego de las 30,000 iteraciones. La figura 5.7 nos muestra el Mapa Topográfico Resultante.



**Figura 5.7 Mapa Topográfico Resultante**

### 5.3.3 Medida de la performance y mecanismos de rechazo

Esta parte no se ha desarrollado en el presente trabajo. Constituye un siguiente paso en la evaluación del modelo, una vez lograda su estabilización.

La medida de la performance nos indica como es que el modelo se comporta en el reconocimiento de dígitos en general (los utilizados en el entrenamiento y otros); esto es necesario como parte de las pruebas para que el producto se ponga a disposición de una manera comercial.

El mecanismo de rechazo que se podría implementar permite efectuar una selección más elaborada, es decir, que acciones tomar si es que los dígitos a reconocer no cumplen con los patrones del Mapa Topológico establecidos. En los dígitos podríamos tener el caso de que si un dígito no es reconocido intentar con el que más se acerca o simplemente descartarlo para su verificación manual. Las implementaciones dependen del uso que se le pretenda dar al

modelo, si por ejemplo se tratara de clasificación postal el mecanismo de rechazo nos debe asegurar que una carta no se vaya a un destino equivocado por los costos que significarían corregir el error.

#### **5.3.4 Aspectos relevantes en la mejora de solución inicial**

- 1) La implementación en las máquinas lectoras de cheques se basa en el empleo de curvas características que guardan la información de cada uno de los tipos, la relación entre tipo y curva características es de uno a uno; se asume que dicho patrón es universal. Esto permitió resolver el problema aunque de una manera rígida pues solo reconocía los tipos preimpresos en tinta magnética. Nuestro modelo generaliza la solución al problema, inclusive nos libera del lector magnético por una implementación vía software para la captura de la información de los dígitos sin la necesidad de que éstos esten escritos con tinta magnética.
- 2) Dichas máquinas sólo pueden reconocer los tipos que tengan la forma establecida de antemano. Los dígitos manuscritos están en función a la persona que los realiza, de ahí que se tengan no uno sino varios patrones; condición que queda sin efecto en nuestro modelo.
- 3) La máquina lectora de cheques utiliza una curva característica para cada tipo, lo cual resta generalidad a la implementación. El modelo planteado permite varios vectores característicos por cada dígito cubriendo de esta forma el amplio espectro de dígitos encontrados en la realidad; el proceso de reconocimiento no es fijo sino que está basado en similitudes o características similares.
- 4) La implementación llevada a cabo en la máquina lectora de cheques almacena 7 valores para cada tipo. El modelo planteado permite hasta 15

lecturas con lo que se obtiene mayor información para cada dígito, asimismo el proceso de captura de la información de los dígitos es de fácil modificación y adaptación de acuerdo a las necesidades de la implementación a realizar además de ser una solución menos costosa.

- 5) El conocimiento acerca de los dígitos es rígido en las máquinas lectoras de cheques. En el modelo implementado este conocimiento puede ser fácilmente alterado lo que permite una solución dinámica al problema, por ejemplo, podríamos utilizar este modelo para reconocer dígitos preimpresos y en determinado formato de una manera eficiente, dígitos escritos por una persona para lograr mayor rendimiento en el reconocimiento, etc. Con ello el espectro de problemas a cubrir es mucho mayor.
- 6) Una vez concluido el entrenamiento, los vectores de características que representen dígitos similares estarán formando espacios definidos en todo el Mapa de Kohonen, esta disposición topológica es lo que nos va a permitir llevar a cabo la tarea del reconocimiento de los dígitos. En el mapa de Kohonen cada neurona almacena la información que haya podido aprender durante su entrenamiento; adicionalmente las neuronas vecinas almacenan información similar, es decir, la información de dígitos con características similares.
- 7) El Mapa Topográfico queda particionado en áreas donde se concentran los dígitos con características similares; el matching que ocurre en las máquinas lectoras de cheques se convierte en un proceso de decisión mediante el cual el dígito “desconocido” buscará ubicarse en el mapa en la región donde su vector de características sea lo más parecido, dicho de otra forma: se ubicará en la región donde el error producto de la

comparación entre su vector de características y los valores almacenados en el Mapa sea mínimo.

- 8) Luego del aprendizaje, cada neurona no almacena los valores característicos de los dígitos usados en el proceso de aprendizaje sino una parte muy importante de ellos; se asume que cada neurona almacena la información de determinado dígito por la similitud de información que almacena, esto nos permite observar gráficamente la formación de las regiones antes mencionadas. Existirán neuronas a las cuales no se les ha definido el dígito que han aprendido a reconocer pero conservan información relativa a los mismos y formarán vecindad con dígitos a los cuales si se les ha definido el dígito que reconocerán.
- 9) Durante el proceso de reconocimiento puede presentarse el caso que el dígito “desconocido” elige a una neurona cuyo dígito no ha sido definido; existen dos alternativas a tomar: la primera es tomar el mayor promedio de la vecindad, es decir, contabilizar todas las repeticiones de todos los dígitos de la vecindad elegida y establecer que el dígito “desconocido” es aquel que tenga la mayor cantidad de repeticiones; la segunda consiste en rechazar el reconocimiento del dígito, es decir, asumir que el dígito “desconocido” no puede ser clasificado.
- 10) La decisión a tomar al respecto depende del tipo de implementación que se quiera realizar, por ejemplo si fuera clasificación postal, lo recomendable sería tomar la segunda alternativa que significaría menor costo ya que un proceso manual podría ayudar a clasificar adecuadamente lo que el modelo no pudo hacer de manera automática, de lo contrario estaríamos enfrentando el problema de por ejemplo clasificar una carta como Lima 1 siendo en realidad Lima 7 con los consiguientes problemas que ello conlleva.

### 5.3.5 Conclusiones

- 1) De acuerdo al nivel de eficiencia obtenido (95%) concluimos que la estabilización del modelo se ha logrado con un margen de error aceptable (5%).
- 2) El modelo elegido nos permite tener un espacio topológicamente “ordenado”, donde encontramos regiones donde se concentran las muestras de un mismo dígito. El Clustering es entonces llevado a cabo de una manera automática por el modelo, lo que nos sugiere que la técnica empleada es útil para resolver los problemas de Clustering.
- 3) Inicialmente pensamos que las regiones deberían ser únicas pero luego de las diferentes corridas del programa pudimos distinguir que se forma no una sino varias regiones, esto se debe a que los dígitos escritos a mano tienen muchas diferencias entre sí lo que se ve con la aparición de varias regiones para un mismo dígito, esta particularidad nos permite que el modelo sea más general.
- 4) El proceso de reconocimiento es ahora un proceso de decisión en el cual se busca que el dígito a reconocer sea ubicado en algún punto de todo el espacio para posteriormente identificar a que región pertenece y por asociación determinar de que dígito se trata.
- 5) Un proceso adicional de afinamiento del modelo lo constituye el aplicar las reglas del LVQ, no desarrollamos esta parte pues los resultados obtenidos son aceptables. Se deja la aplicación de las reglas mencionadas para mejoras al trabajo efectuado.



### **5.3.6 Limitaciones del modelo**

- 1) El éxito del modelo radica en una buena elección de los vectores de características, por lo que la principal limitación se encuentra en la definición de las mismas las cuales como mencionamos anteriormente no tienen reglas precisas de obtención.
- 2) Para el aprendizaje es necesario contar con computadoras de gran potencia de cálculo, el número de neuronas que se utilizó si bien es cierto nos ha permitido obtener resultados aceptables, no pudo verse incrementado pues el proceso de entrenamiento iba a tomar demasiado tiempo considerando la cantidad de pruebas previas a realizar. Este tipo de desarrollo exigen el uso de máquinas con gran poder de procesamiento.
- 3) El modelo asume que los dígitos a reconocer no tienen (o son mínimos) marcas extrañas; para ellos es necesario la implementación de técnicas de filtrado de información que corresponde a la parte del tratamiento de imágenes.
- 4) La escritura de los dígitos no es libre, es necesario el uso de casilleros similares a los encontrados en los formatos preimpresos por lo que el espectro de implementaciones puede verse reducido.

### **5.4 Conclusiones**

Es factible el uso de las redes neuronales artificiales para resolver problemas donde técnicas tradicionales no brindan buenos resultados.

## **Capítulo 6 Consideraciones Finales**

### **6.1 Recomendaciones**

La implementación del modelo no solamente es válido para el problema planteado; algunos componentes y sobre todo la técnica utilizado son factibles de ser reusados para otros problemas relacionados con el reconocimiento de patrones.

La rutina de lectura de los archivos BMP, por ejemplo, ofrece perspectivas para implementaciones relacionadas con el tratamiento de imágenes.

La implementación del PERCEPTRON puede ser utilizado para problemas de clasificación tipo IF-THEN-ELSE.

Consideramos que lo más importante es haber descrito una implementación mediante el uso de la tecnología de las redes neuronales artificiales, lo cual esperamos sea punto de partida para el inicio en el desarrollo de aplicaciones basadas en ella.

A continuación se describe una implementación mediante el modelo desarrollado.

### **6.2 Diseño de un Clasificador**

En la Figura 6.1 se muestra el esquema del Clasificador Postal Propuesto para ser utilizado dentro de la ciudad de Lima.

El diseño del sobre es mostrado en la parte superior; en la parte del remitente se han dispuesto dos recuadros para indicar el código del distrito al igual que en la parte del remitente. Toda la información, como es de suponer, es llenada a mano.

A fin de calibrar adecuadamente el scanner dichas zonas deberán permanecer fijas.

Se ha dispuesto una Banda Sinfin por donde circularán las cartas. El Clasificador Postal detectará la presencia de una carta y procederá a capturar la imagen del mismo; el módulo encargado del reconocimiento determinará el distrito del Destinatario y enviará una orden al controlador mecánico de compuertas que actuará en función a la orden recibida por el Clasificador Postal. Las cartas circulan de

manera secuencial y el surtidor de las mismas deberá funcionar a una velocidad determinada por el tiempo requerido para clasificar una carta.

**Cálculo del tiempo requerido para Clasificar una carta:**

**T:** Tiempo requerido para Clasificar una carta.

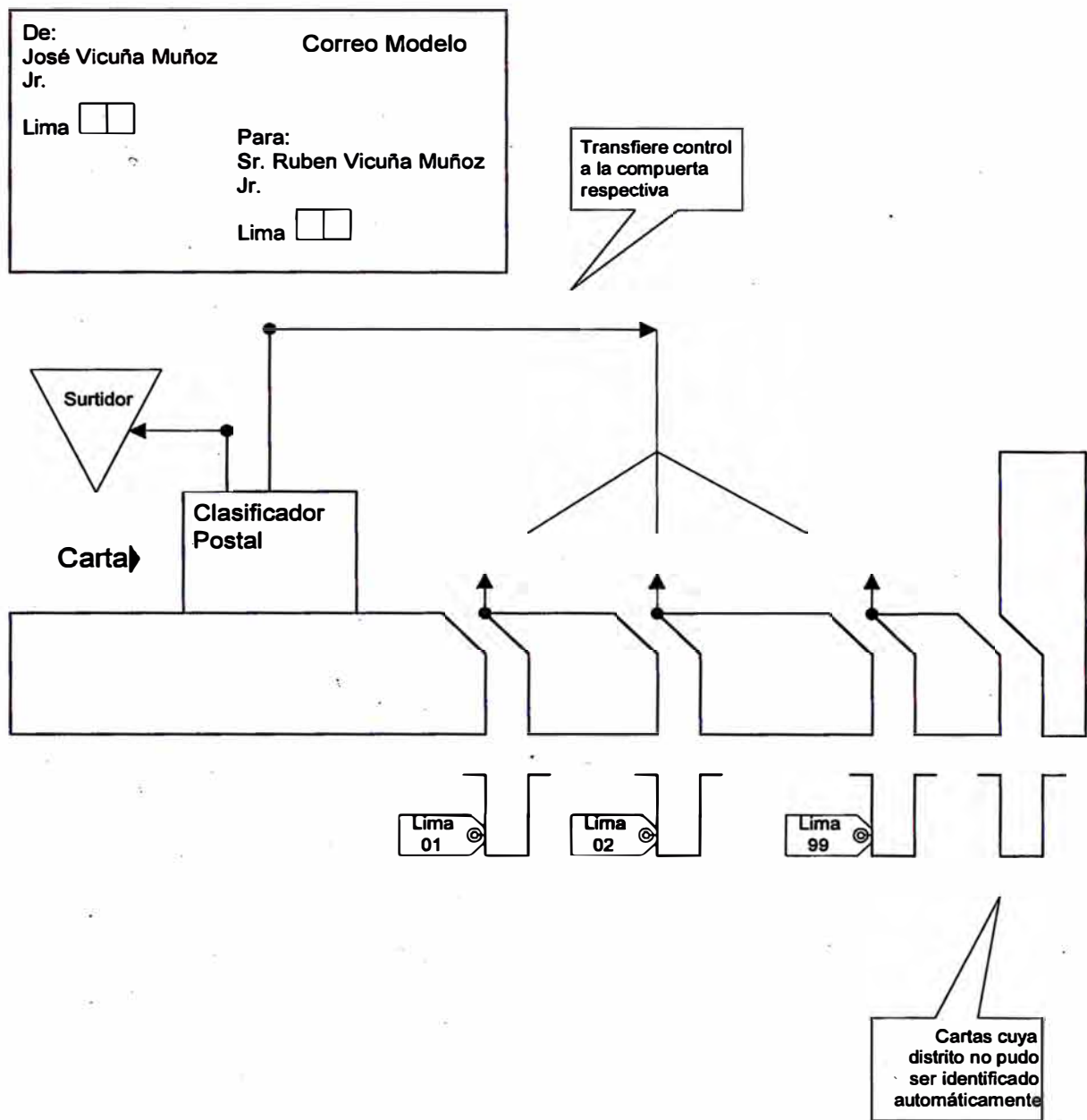
**Td:** Tiempo requerido por el Clasificador para determinar el Distrito del Destinatario

**Tc:** Tiempo requerido para que el Controlador active la compuerta correspondiente una vez recibida la orden del Clasificador Postal.

**Tb:** Tiempo requerido para que la carta sea depositado en el buzón correspondiente al distrito del Destinatario.

$$\mathbf{T = Td + Tc + Tb}$$

Es posible acortar el tiempo **T** incorporándole al Controlador de Compuertas la capacidad de controlar el flujo de más de una carta, para lo cual dicho controlador deberá indicar al surtidor el momento en el cual enviar una siguiente carta.



**Figura 6.1 Diagrama de un Clasificador**

El mismo esquema es válido para el proceso de vouchers de las tarjetas de crédito.

## ANEXO 1

### Lista de valores utilizados para el entrenamiento del Perceptron

Muestra	x1	x2	Valor Deseado para el 7	x1	x2	Valor Deseado para el 1
1	0.6667	0.9431	-1	1.5111	2.0456	1
2	0.6857	0.7835	-1	1.1333	2.1648	1
3	0.7407	0.8432	-1	1.8601	1.8569	1
4	0.7556	0.9444	-1	1.6845	1.1005	1
5	0.7714	1.3500	-1	2.0000	1.3012	1
6	0.8000	1.0060	-1	1.4286	2.5401	1
7	0.8000	0.7955	-1	1.1026	1.9277	1
8	0.8000	0.7874	-1	2.1154	1.9082	1
9	0.8000	0.7874	-1	1.8846	4.4375	1
10	0.8400	1.2358	-1	1.8413	1.8321	1
11	0.8750	0.7621	-1	1.3333	2.2044	1
12	0.9000	0.7961	-1	1.4286	3.4821	1
13	0.9231	1.6507	-1	1.7000	1.3288	1
14	0.9259	1.6005	-1	1.8308	2.5642	1
15	0.9259	0.7275	-1	1.4103	1.8608	1
16	0.9259	1.1936	-1	2.1154	1.7843	1
17	0.9444	1.4000	-1	2.0357	1.3170	1
18	0.9643	0.9757	-1	1.7104	2.7320	1
19	1.0000	1.7932	-1	2.0211	2.4692	1
20	1.0000	1.1048	-1	1.8182	4.1355	1
21	1.0000	1.6406	-1	1.3636	4.3939	1
22	1.0000	2.3333	-1	1.9378	3.5414	1
23	1.0000	2.3333	-1	1.7647	3.3670	1
24	1.0000	1.0000	-1	1.4286	1.5854	1
25	1.0417	1.5044	-1	1.4286	2.6571	1
26	1.1556	0.7797	-1	2.0301	2.9646	1
27	1.2000	1.9297	-1	1.4286	2.7857	1
28	1.2000	2.0351	-1	1.8583	3.0954	1
29	1.2000	1.8828	-1	1.7104	1.2148	1
30	1.5714	1.2107	-1	2.0938	2.5717	1

## ANEXO 2

### Implementación del Algoritmo del Perceptron

```
/*-----*/
/* NEURAL NETWORK V 1.0 Copyright © */
/* */
/* PERCEPTRON */
/* */
/* autor          Vicuña Muñoz, José Antonio */
/* fecha          10-nov-93 */
/* descripción    Implementación del algoritmo */
/*-----*/

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <graphics.h>

#define MAXTIME 300 /* tiempo para el aprendizaje */
#define MAXDATA 60 /* numero de ejemplos para el aprendizaje */

/* Begin main function */

int main(void)
{
    /* request auto detection */
    int gdriver = DETECT, gmode, errorcode;
    int maxx, maxy, color;

    float x[2][MAXDATA];          /* input vectors */
    float d[MAXDATA];            /* desired values */
    float r[MAXDATA];            /* ocurrencias */
    float w0,w1,w2;              /* ponderaciones */
    float etha = 0.1;
    float y, z, wx, x1, x2;
    int seed = 10000;
    int t,i,j,cx;

    /*----- inicializa los valores deseados -----*/
    for ( i = 0; i < 60 ; i++) {
```

```

    if (i < 30)
        d[i] = 1.0;
    else
        d[i] = -1.0;
    r[i] = 0;
}

/*----- lectura de los valores para entrenamiento -----*/
i = 0;
while (scanf("%f %f",&x1,&x2) !=EOF) {
    if (i<60) {
        x[0][i] = x1;
        x[1][i] = x2;
        printf("%9.4f %9.4f\n",x[0][i],x[0][i]);
        i = i + 1;
    }
}
/*----- inicializacion de los valores aleatorios -----*/
srand( seed );
wx = random(10000);
w1 = wx/10000;
wx = random(10000);
w2 = wx/10000;
wx = random(10000);
w0 = -wx/10000;

/*----- lista los valores para el entrenamiento -----*/
printf("INPUT DATA MATRIX\n");
printf("-----\n");
for ( i = 0; i < MAXDATA ; i++) {
    printf("%6d %9.5f %9.5f %5.0f\n",i,x[0][i],x[1][i],d[i]);
}
printf(" initial WEIGHTS\n");
printf(" -----\n");
printf(" w0---> %9.5f\n",w0);
printf(" w1---> %9.5f\n",w1);
printf(" w2---> %9.5f\n",w2);

/*----- proceso de aprendizaje -----*/
for ( t = 0; t < MAXTIME; t++) {
    cx = random(MAXDATA);
    r[cx] = r[cx] + 1;
    y = w1*x[0][cx] + w2*x[1][cx] - w0;
    if (y >= 0.0)
        z = 1.0;
}

```

```

else
    z = -1.0;
    w1 = w1 + etha*(d[cx]-z)*x[0][cx];
    w2 = w2 + etha*(d[cx]-z)*x[1][cx];
}
/*----- lista los valores luego del aprendizaje -----*/
printf(" FINAL WEIGHTS\n");
printf(" -----\n");
printf(" w0 ---> %8.5f\n",w0);
printf(" w1 ---> %8.5f\n",w1);
printf(" w2 ---> %8.5f\n",w2);
printf(" -w1/w2-> %8.5f\n",-w1/w2);
printf(" w0/w2--> %8.5f\n",w0/w2);

/*----- inicializa el modo grafico -----*/
initgraph(&gdriver, &gmode, "");
errorcode = graphresult();
if (errorcode != grOk) {
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1);
}
/*----- imprime el grafico en la pantalla -----*/
maxx = getmaxx() + 1;
maxy = getmaxy() + 1;
color = getmaxcolor() + 1;
setcolor(5);
rectangle( 0, 0, maxx-1,maxy-1);
rectangle(50,30,50+180,30+240);
for (i=0;i<60;i++) {
    if (i < 30) {
        putpixel(50+x[0][i]*50,30+x[1][i]*50,2);
        circle(50+x[0][i]*50,30+x[1][i]*50,2);
    }
    else {
        putpixel(50+x[0][i]*50,30+x[1][i]*50-2,2);
        putpixel(50+x[0][i]*50,30+x[1][i]*50-1,2);
        putpixel(50+x[0][i]*50,30+x[1][i]*50,2);
        putpixel(50+x[0][i]*50,30+x[1][i]*50+1,2);
        putpixel(50+x[0][i]*50,30+x[1][i]*50+2,2);
        putpixel(50+x[0][i]*50-2,30+x[1][i]*50,2);
        putpixel(50+x[0][i]*50-1,30+x[1][i]*50,2);
        putpixel(50+x[0][i]*50+1,30+x[1][i]*50,2);
        putpixel(50+x[0][i]*50+2,30+x[1][i]*50,2);
    }
}

```



```
    }  
  }  
  line(50,30.0+(w0/w2)*50.0,50.0+(w0/w1)*50.0,30);  
  getch();  
  closegraph();  
  return(0);  
}
```

### ANEXO 3

#### Pesos Finales luego del entrenamiento del modelo Perceptron

Muestra	X1	X2	Muestra	X1	X2
1	1.51110	2.04560	16	0.80000	1.00600
2	1.13330	2.16480	17	0.80000	1.00600
3	1.86010	1.85690	18	1.57140	1.21070
4	1.68450	1.10050	19	1.15560	0.77970
5	2.00000	1.30120	20	0.90000	0.79610
6	1.42860	2.54010	21	1.20000	1.92970
7	1.10260	1.92770	22	0.75560	0.94440
8	2.11540	1.90820	23	1.42860	1.74360
9	1.88460	2.43750	24	0.80000	0.79550
10	1.84130	1.83210	25	1.20000	2.03510
11	1.66670	2.58330	26	0.68570	0.78350
12	1.42860	3.48210	27	0.94440	1.40000
13	1.70000	1.32880	28	0.92310	1.65070
14	1.83080	2.56420	29	0.82110	1.00550
15	1.15140	2.10140	30	0.81230	1.01000

## ANEXO 4

### Vectores de Características utilizados en el entrenamiento

0	23.57	17.86	11.43	10.72	9.29	12.15	12.86	15.72	27.14	4.29	0.00	0.00	0.00	0.00	0.00
0	31.58	19.08	10.53	8.55	7.24	9.87	17.77	40.13	5.92	0.00	0.00	0.00	0.00	0.00	0.00
0	26.92	13.46	8.33	7.05	6.41	9.62	15.38	32.05	24.36	0.00	0.00	0.00	0.00	0.00	0.00
0	30.41	16.22	12.16	10.14	10.14	14.87	19.60	33.11	6.08	0.00	0.00	0.00	0.00	0.00	0.00
0	24.39	14.02	7.93	9.15	10.37	18.90	34.76	23.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	24.41	13.10	10.12	8.33	8.33	15.48	35.12	19.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	33.33	20.51	12.18	9.62	8.97	14.10	15.38	28.21	17.95	0.00	0.00	0.00	0.00	0.00	0.00
0	25.00	12.50	9.09	9.09	7.96	9.09	12.50	37.50	11.36	0.00	0.00	0.00	0.00	0.00	0.00
0	22.80	16.92	12.50	11.76	12.50	10.29	13.24	28.68	19.12	0.00	0.00	0.00	0.00	0.00	0.00
0	27.70	14.87	10.81	11.49	13.51	18.24	27.03	4.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	39.21	21.59	10.80	7.96	18.75	31.82	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	23.78	12.20	9.76	10.37	8.54	12.20	17.07	25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	30.49	17.68	10.98	9.76	10.37	20.12	36.59	6.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	29.41	14.71	11.76	9.56	10.29	14.71	17.65	30.15	6.62	0.00	0.00	0.00	0.00	0.00	0.00
0	27.86	15.00	10.00	9.29	13.57	38.57	10.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	29.76	16.67	10.12	10.71	13.10	14.29	26.79	22.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	32.43	14.86	13.51	8.79	8.11	13.51	20.27	23.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	25.00	16.67	8.98	8.33	12.82	27.56	25.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	30.77	14.74	10.90	10.26	8.98	15.39	31.41	15.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	25.00	15.15	10.61	11.36	15.91	34.85	25.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	37.50	50.00	20.00	3.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	7.78	11.67	30.00	37.22	24.44	23.34	3.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	11.05	13.37	12.79	23.84	26.16	31.98	19.19	4.07	5.81	4.07	0.00	0.00	0.00	0.00	0.00
1	16.67	27.78	25.56	16.12	8.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	7.96	36.36	71.02	1.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	8.17	8.66	9.62	35.10	38.46	27.40	8.66	4.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	6.55	17.26	61.31	36.91	19.65	4.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	8.53	21.03	52.84	31.25	19.89	6.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

1	5.00	6.67	6.67	7.78	45.56	44.44	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	26.02	35.21	20.92	17.35	7.65	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	13.10	15.48	38.69	55.95	17.26	6.55	1.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	16.45	26.97	21.05	15.13	3.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	22.97	22.97	22.30	5.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	5.12	6.82	7.39	25.00	38.07	20.45	15.34	6.82	6.25	3.41	0.00	0.00	0.00	0.00	0.00
1	6.67	8.33	21.11	28.33	26.67	17.78	6.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	11.11	33.34	31.95	23.61	9.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	4.55	4.55	3.41	3.41	23.30	30.12	21.03	4.55	3.41	0.00	0.00	0.00	0.00	0.00	0.00
1	6.82	18.75	39.77	34.66	15.91	7.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	3.41	3.98	11.93	35.80	45.46	19.32	3.98	4.55	2.28	0.00	0.00	0.00	0.00	0.00	0.00
1	5.32	7.45	10.11	29.79	32.45	27.13	16.49	6.39	1.60	0.00	0.00	0.00	0.00	0.00	0.00
2	10.65	13.43	11.57	16.20	18.52	19.45	25.00	7.87	6.95	2.32	0.00	0.00	0.00	0.00	0.00
2	21.64	22.60	24.52	26.93	35.58	32.69	28.85	5.77	4.33	5.77	1.92	0.00	0.00	0.00	0.00
2	9.46	20.27	19.60	19.60	18.92	20.95	10.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	11.54	21.15	23.72	21.16	19.23	23.72	11.54	4.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	37.24	30.61	26.53	28.06	34.69	38.27	10.71	10.20	11.23	0.00	0.00	0.00	0.00	0.00	0.00
2	10.80	21.02	23.87	22.16	18.75	22.16	24.43	10.23	0.57	0.00	0.00	0.00	0.00	0.00	0.00
2	16.67	23.89	20.56	19.45	17.23	20.56	8.89	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	12.25	16.84	12.25	19.39	23.47	20.41	22.96	23.98	3.06	0.00	0.00	0.00	0.00	0.00	0.00
2	23.03	19.74	17.76	17.76	23.69	9.87	10.53	3.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	20.11	28.81	18.48	14.67	14.13	14.67	13.04	21.74	20.11	10.33	0.00	0.00	0.00	0.00	0.00
2	8.89	16.67	14.45	20.00	16.67	13.33	15.00	25.56	14.45	2.22	0.00	0.00	0.00	0.00	0.00
2	11.88	15.00	12.50	28.75	31.25	32.50	30.00	17.50	8.13	8.13	0.00	0.00	0.00	0.00	0.00
2	32.39	32.96	21.02	20.45	27.84	26.14	2.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	16.28	14.54	14.54	25.58	25.58	32.56	36.63	18.02	9.88	0.00	0.00	0.00	0.00	0.00	0.00
2	8.33	10.26	12.82	18.59	19.23	25.00	14.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	18.90	17.68	24.39	26.83	32.93	28.66	26.22	18.30	5.49	0.00	0.00	0.00	0.00	0.00	0.00
2	14.68	13.59	18.48	23.37	36.96	11.41	7.07	15.22	1.63	0.00	0.00	0.00	0.00	0.00	0.00
2	22.22	15.56	15.56	22.78	24.44	27.78	31.11	11.11	7.23	2.78	0.00	0.00	0.00	0.00	0.00
2	32.39	32.96	21.02	20.45	27.84	26.14	2.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

2	16.28	14.54	14.54	25.58	25.58	32.56	36.63	18.02	9.88	0.00	0.00	0.00	0.00	0.00	0.00
3	19.77	23.26	19.77	23.84	30.82	31.98	6.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	14.19	16.22	21.62	29.05	29.06	27.03	25.68	18.24	19.60	6.76	0.00	0.00	0.00	0.00	0.00
3	9.05	10.11	17.02	20.22	21.28	22.34	31.38	13.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	10.87	14.13	17.94	23.91	21.74	20.11	25.54	15.76	15.22	2.72	0.00	0.00	0.00	0.00	0.00
3	9.76	18.90	20.73	24.39	21.95	23.78	25.61	22.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	14.64	24.39	25.00	23.78	31.10	43.90	11.59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	13.89	17.78	21.67	22.22	23.33	21.67	14.45	16.67	5.00	0.00	0.00	0.00	0.00	0.00	0.00
3	17.22	10.56	7.78	13.34	16.67	21.11	20.56	15.56	18.89	28.34	11.67	0.00	0.00	0.00	0.00
3	11.54	7.05	8.34	17.31	18.59	23.08	28.85	26.93	10.26	0.00	0.00	0.00	0.00	0.00	0.00
3	10.78	18.63	18.14	17.65	19.12	26.96	39.22	5.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	12.50	11.90	12.50	17.27	20.24	20.84	18.46	23.81	25.60	1.79	0.00	0.00	0.00	0.00	0.00
3	12.50	10.53	7.89	19.74	21.05	19.74	24.34	32.90	19.08	0.00	0.00	0.00	0.00	0.00	0.00
3	6.11	6.67	8.34	14.45	16.11	16.11	14.45	15.56	22.78	27.22	2.78	0.00	0.00	0.00	0.00
3	7.32	11.59	14.03	20.73	22.56	22.56	14.03	15.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	12.50	15.13	17.11	19.74	24.34	28.95	32.24	28.29	1.97	0.00	0.00	0.00	0.00	0.00	0.00
3	5.95	5.36	10.71	15.48	17.27	17.27	16.67	16.08	15.48	19.65	17.26	5.95	0.00	0.00	0.00
3	8.89	11.67	25.56	24.44	19.45	25.56	32.78	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	9.38	8.13	7.50	13.13	18.13	18.75	24.38	30.00	19.38	10.63	0.63	0.00	0.00	0.00	0.00
3	5.63	6.88	10.00	11.25	12.50	16.25	23.13	25.00	23.13	22.50	10.63	0.00	0.00	0.00	0.00
3	10.37	9.76	15.86	26.22	28.05	24.39	24.39	29.27	9.15	0.00	0.00	0.00	0.00	0.00	0.00
4	9.66	15.34	15.91	15.34	47.73	44.32	1.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	34.72	15.97	8.33	8.33	64.58	19.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	39.59	11.80	7.64	28.48	56.94	14.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	9.29	12.86	11.43	12.86	12.14	35.00	45.72	2.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	30.71	8.57	5.71	57.86	23.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	27.14	23.57	6.43	8.57	38.57	52.86	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	5.82	11.63	12.79	12.21	13.95	34.30	41.86	0.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	8.33	23.72	13.46	3.85	5.13	4.49	55.13	19.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	12.86	15.72	13.58	13.57	15.00	47.86	51.43	2.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	8.79	16.89	29.73	9.46	4.06	16.22	54.06	12.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00

4	11.18	23.03	13.82	2.63	15.79	68.42	9.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	17.05	19.89	9.09	19.89	47.73	15.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	22.92	19.45	5.56	5.56	14.59	66.67	8.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	7.74	12.50	12.50	12.50	13.69	30.95	50.00	5.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	9.61	11.54	10.90	11.54	47.44	33.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	8.70	13.04	12.50	12.50	5.43	26.63	32.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	9.76	22.56	19.51	4.88	40.86	42.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	14.10	16.02	15.39	12.18	21.79	69.87	12.82	7.69	1.92	0.00	0.00	0.00	0.00	0.00	0.00
4	9.31	21.08	22.06	20.59	39.71	73.04	18.63	5.88	0.49	0.00	0.00	0.00	0.00	0.00	0.00
4	11.54	14.10	14.74	16.03	39.74	38.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	24.24	30.30	19.70	16.67	18.94	28.03	19.70	3.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10.47	6.40	22.09	23.25	19.18	15.12	17.44	27.91	7.56	0.00	0.00	0.00	0.00	0.00	0.00
5	33.78	22.30	22.30	21.62	21.62	26.35	27.03	2.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	33.78	22.30	22.30	21.62	21.62	26.35	27.03	2.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	14.10	27.57	19.87	19.23	18.59	23.08	11.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	12.50	26.47	10.29	10.29	18.39	37.50	16.18	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	18.55	27.42	24.19	23.39	25.00	31.45	33.07	2.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	27.03	21.62	19.60	18.92	23.65	22.97	10.81	6.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	14.84	32.03	14.84	18.75	26.56	40.63	13.28	3.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	18.38	31.62	16.92	17.65	19.12	29.41	16.18	7.35	5.15	0.00	0.00	0.00	0.00	0.00	0.00
5	29.69	26.57	19.53	23.44	34.38	25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	7.26	15.32	27.42	18.55	17.74	24.19	31.45	10.49	7.26	0.00	0.00	0.00	0.00	0.00	0.00
5	19.17	35.83	22.50	30.00	7.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	3.84	21.79	33.33	14.74	20.51	21.15	1.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	9.17	5.84	16.67	27.50	22.50	23.34	23.33	7.50	7.50	0.83	0.00	0.00	0.00	0.00	0.00
5	22.86	23.57	20.00	21.43	21.43	26.43	9.29	1.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	26.39	26.39	13.89	17.36	23.61	37.50	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	16.89	14.87	25.67	16.89	16.22	23.65	32.44	7.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5.15	6.62	25.74	19.12	14.71	22.80	25.00	7.35	8.09	1.47	0.00	0.00	0.00	0.00	0.00
5	15.79	29.60	16.45	16.45	18.42	25.66	9.21	7.89	3.29	0.00	0.00	0.00	0.00	0.00	0.00
6	51.70	33.52	15.34	9.66	15.91	5.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

6	48.91	24.46	17.94	19.02	28.26	8.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	29.79	29.26	21.81	14.89	12.77	5.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	33.11	30.41	25.00	19.59	29.05	13.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	47.62	30.95	19.05	22.62	19.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	43.75	32.50	22.50	16.25	20.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	38.16	28.29	21.05	16.45	13.82	9.87	14.47	1.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	30.47	24.22	25.00	17.97	17.19	10.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	40.25	29.88	26.22	17.68	14.02	11.59	15.25	2.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	42.78	32.78	22.78	18.89	16.11	17.22	2.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	39.64	37.81	25.61	12.20	8.54	16.46	3.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	31.67	22.78	18.34	16.67	16.67	27.78	6.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	45.73	26.22	18.29	15.24	16.46	14.64	1.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	58.34	46.67	28.89	25.00	38.89	21.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	38.13	25.00	16.88	10.00	13.75	8.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	43.24	30.41	18.92	14.19	10.14	14.19	3.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	52.50	35.63	23.13	16.88	20.63	4.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	43.02	31.98	21.51	16.28	18.02	10.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	36.46	27.60	17.71	11.98	12.50	1.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	22.22	36.67	19.45	12.78	9.45	9.45	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	5.92	6.58	17.11	57.90	39.47	5.92	6.58	2.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	14.13	4.89	6.52	34.24	30.44	16.85	2.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	7.35	27.94	34.56	36.77	15.44	5.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	10.16	10.94	35.94	43.75	16.41	5.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	24.27	18.38	27.94	52.94	30.88	8.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	11.81	7.64	18.75	34.03	27.78	20.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	18.05	36.11	20.83	18.06	17.36	6.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	8.82	21.32	37.50	37.50	18.38	2.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	11.18	44.08	28.95	22.37	20.39	13.82	3.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	12.81	4.88	13.42	38.42	29.88	23.78	3.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	8.82	29.42	36.03	38.24	16.91	1.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	13.75	9.38	31.88	29.38	26.88	3.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

7	16.91	5.15	13.97	41.18	32.35	2.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	8.13	23.13	46.88	34.38	6.25	2.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	7.69	10.26	41.03	42.31	14.74	8.33	2.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	13.10	12.50	13.10	33.93	33.33	24.41	6.55	1.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	4.86	11.81	38.20	56.94	15.28	4.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	4.17	7.64	29.17	39.58	37.50	13.19	6.95	1.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	4.55	21.21	31.06	34.85	9.85	7.58	3.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	25.00	21.43	15.72	14.29	15.00	16.43	5.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	12.81	15.86	25.61	20.73	20.12	26.83	26.83	4.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	25.00	28.57	40.71	19.29	16.43	16.43	23.57	28.57	4.29	0.00	0.00	0.00	0.00	0.00	0.00
8	33.93	24.41	19.05	20.84	31.55	13.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	39.59	34.03	29.17	47.92	28.47	1.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	36.88	38.75	20.00	16.25	15.00	17.50	24.38	21.88	8.75	0.00	0.00	0.00	0.00	0.00	0.00
8	39.54	29.65	27.33	32.56	29.65	2.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	18.75	13.13	19.38	30.63	30.00	34.38	6.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	34.30	36.05	28.49	15.70	11.05	15.70	2.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	37.84	50.68	29.05	19.60	18.25	27.03	22.97	19.59	10.14	0.00	0.00	0.00	0.00	0.00	0.00
8	30.00	24.44	22.78	28.89	27.78	11.11	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	18.18	19.70	33.33	31.06	21.97	21.97	22.73	30.30	31.06	2.27	0.00	0.00	0.00	0.00	0.00
8	39.03	26.22	21.95	15.85	20.12	21.34	10.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	28.68	41.18	33.82	42.65	38.97	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	36.84	28.95	26.32	31.58	28.29	13.16	3.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	50.00	28.79	19.70	18.94	22.73	37.12	24.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	41.47	29.27	18.90	18.29	23.78	40.86	34.15	4.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	15.70	14.53	25.58	30.82	30.23	38.95	23.26	2.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	26.74	37.21	27.91	19.77	20.93	23.26	26.16	16.86	9.88	0.00	0.00	0.00	0.00	0.00	0.00
8	37.50	35.12	30.36	27.38	33.33	14.88	9.52	2.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	21.62	23.65	33.11	21.62	18.25	17.57	23.65	26.35	1.35	0.00	0.00	0.00	0.00	0.00	0.00
9	23.08	12.82	26.28	55.77	17.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	19.79	15.10	13.02	16.15	36.98	33.86	1.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	21.28	11.17	13.30	18.62	50.00	38.30	9.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00



9	19.65	17.86	33.33	22.62	25.00	21.43	0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	20.93	11.63	10.47	31.40	42.44	33.14	4.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	20.93	11.63	10.47	16.86	62.21	15.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	18.45	11.31	24.41	38.10	28.57	15.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	17.39	15.22	36.96	32.07	24.46	9.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	22.22	20.00	12.22	25.00	40.00	27.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	26.07	19.15	18.09	29.79	22.34	0.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	10.55	8.60	14.46	27.74	20.70	14.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	23.89	17.23	22.78	23.89	25.56	23.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	25.00	13.75	11.88	21.25	58.13	11.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	21.20	10.87	10.87	20.65	66.30	7.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	19.89	9.09	11.36	14.77	50.57	22.16	2.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	23.22	16.08	28.57	33.93	33.33	15.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	25.60	17.27	20.24	26.19	55.95	17.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	18.48	11.96	14.67	57.61	30.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	15.91	12.50	11.93	10.79	52.84	32.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	18.48	14.13	7.61	8.70	30.98	39.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

## ANEXO 5

### Valores Finales de los pesos luego del entrenamiento del SOFM

0	28.64078	13.70396	11.36677	20.19612	60.27838	14.66769	0.04198	0.00042	0.00021	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	25.82493	13.85806	11.14829	19.50115	57.38026	15.74287	0.33057	0.00004	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	18.04835	13.84934	12.14194	16.36978	49.13235	25.38507	0.84433	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	14.47744	11.55437	11.84319	13.80825	46.10686	35.10932	2.33084	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	11.62309	13.00472	12.93183	12.35728	46.18079	39.40614	2.05056	0.00021	0.00015	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	12.94039	15.08587	11.29094	9.95514	40.81342	43.41147	1.07891	0.00272	0.00121	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	17.31821	18.62985	10.80270	7.69302	35.35500	43.49182	1.59783	0.15165	0.09738	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	12.58878	14.40038	7.34872	7.29121	25.25937	36.32958	11.65987	1.63240	1.08203	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	12.17475	14.50316	7.32238	7.28227	20.13443	34.05320	18.87397	2.12711	1.38336	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	9.28949	13.53537	9.38418	9.27716	15.66429	32.38369	27.57041	1.63900	0.92898	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	8.29076	12.59467	12.20293	12.27735	12.16822	32.36708	41.88707	2.04844	0.04187	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
11	8.68220	13.05769	12.67180	12.69604	13.03745	35.43690	45.36323	2.45086	0.00226	0.00006	0.00000	0.00000	0.00000	0.00000	0.00000
12	10.85656	14.16265	15.40287	14.02284	14.53937	34.08611	45.01247	4.39052	0.08287	0.00413	0.00000	0.00000	0.00000	0.00000	0.00000
13	12.41310	15.38372	23.45285	17.79986	17.20989	26.26893	33.12413	6.75300	1.25101	0.01132	0.00000	0.00000	0.00000	0.00000	0.00000
14	11.79976	15.95058	25.41678	19.49587	18.06927	24.38418	29.64253	8.28451	1.94096	0.03266	0.00000	0.00000	0.00000	0.00000	0.00000
15	9.32244	15.73652	24.99433	21.46480	18.58557	23.79452	29.06676	10.09908	2.39120	0.09340	0.00000	0.00000	0.00000	0.00000	0.00000
16	8.75802	12.81453	21.22790	19.97014	18.30262	22.52054	27.85545	9.70107	4.09073	0.80452	0.00000	0.00000	0.00000	0.00000	0.00000
17	8.64889	10.48218	17.83948	19.31269	18.62035	21.48193	25.82536	9.64780	5.68084	1.31183	0.00000	0.00000	0.00000	0.00000	0.00000
18	8.83952	10.62292	17.09873	21.33238	20.02847	21.50882	22.61941	11.11190	7.41690	1.44861	0.00000	0.00000	0.00000	0.00000	0.00000
19	9.44370	10.90388	16.59554	23.63677	21.99441	21.69100	21.18949	13.46585	7.53761	1.16895	0.00000	0.00000	0.00000	0.00000	0.00000
20	27.90474	12.87298	10.72105	22.21996	59.30597	13.68387	0.49790	0.19915	0.01158	0.00129	0.00000	0.00000	0.00000	0.00000	0.00000
21	25.47867	14.09777	11.47402	22.63174	57.24183	14.18386	0.44360	0.17257	0.00317	0.00022	0.00000	0.00000	0.00000	0.00000	0.00000
22	21.48512	15.05985	12.40209	21.53118	48.39809	22.47381	1.09297	0.00050	0.00018	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000
23	17.99178	14.02077	11.76821	18.50926	45.43824	29.92632	2.02762	0.00677	0.00085	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000
24	12.98686	12.28423	12.26753	13.81039	45.53593	38.22941	2.22162	0.01209	0.00134	0.00003	0.00000	0.00000	0.00000	0.00000	0.00000
25	13.90180	15.21014	12.26572	11.89766	43.66430	43.60399	2.26489	0.01988	0.00241	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000

26	16.80998	18.49209	11.34919	8.89097	37.70494	45.58509	0.97080	0.10720	0.02756	0.00014	0.00000	0.00000	0.00000	0.00000	0.00000
27	21.04669	18.16082	7.21645	7.61076	27.05703	51.13972	4.40617	0.25356	0.11140	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
28	15.81342	15.90820	7.06954	6.82726	18.34996	43.67189	14.81493	1.57159	1.00146	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
29	16.52742	14.85823	8.52267	8.37124	16.39324	32.35220	21.67941	1.29664	0.84527	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
30	14.53210	13.93399	11.21994	11.02185	12.37982	31.37024	32.95452	0.42381	0.05141	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
31	8.98649	13.15888	12.63429	12.69580	12.53817	34.89226	44.14236	2.21810	0.00089	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000
32	9.77958	14.55558	13.97119	13.94708	14.92836	34.76775	45.72575	3.70250	0.00738	0.00263	0.00000	0.00000	0.00000	0.00000	0.00000
33	12.99320	16.18242	20.24092	16.94427	17.23097	31.16973	38.01231	5.30792	0.09597	0.00501	0.00000	0.00000	0.00000	0.00000	0.00000
34	12.59257	16.31538	24.68372	19.51168	18.35355	24.83888	30.20889	7.64809	0.99607	0.01980	0.00001	0.00000	0.00000	0.00000	0.00000
35	10.50460	16.00740	25.26893	21.17044	18.84521	24.48573	28.93249	9.02001	1.68480	0.03146	0.00000	0.00000	0.00000	0.00000	0.00000
36	9.56219	14.99790	23.47156	21.62789	19.24287	23.68406	29.14343	10.87964	1.64817	0.10492	0.00000	0.00000	0.00000	0.00000	0.00000
37	8.99224	12.57007	20.12264	20.29324	18.72524	22.43775	27.18087	10.51479	3.69012	0.89865	0.00000	0.00000	0.00000	0.00000	0.00000
38	9.03588	11.85296	17.42507	20.44452	20.03149	21.49337	23.31434	12.56759	6.19462	1.27361	0.00000	0.00000	0.00000	0.00000	0.00000
39	10.29125	12.39440	17.58638	23.42306	22.37070	21.80874	19.91295	14.29271	6.92140	0.92392	0.00000	0.00000	0.00000	0.00000	0.00000
40	18.32598	11.34525	14.95622	32.46156	50.90218	12.15505	2.51311	0.89711	0.06152	0.00534	0.00000	0.00000	0.00000	0.00000	0.00000
41	19.32412	14.94221	12.21130	27.97662	45.29588	20.94390	2.29732	0.47877	0.02967	0.01211	0.00000	0.00000	0.00000	0.00000	0.00000
42	18.68017	15.74763	11.53489	24.67486	40.96561	26.99790	2.47185	0.47726	0.05046	0.01821	0.00000	0.00000	0.00000	0.00000	0.00000
43	18.80070	14.49760	13.37760	22.11027	41.21004	30.30750	3.71089	2.27425	0.29168	0.03416	0.00000	0.00000	0.00000	0.00000	0.00000
44	15.65072	14.33142	14.12931	17.60523	42.57582	34.11416	4.26910	3.48166	0.42724	0.04421	0.00000	0.00000	0.00000	0.00000	0.00000
45	18.35317	18.84355	10.40455	11.39153	37.77505	48.53012	2.72059	0.62360	0.20690	0.08319	0.00000	0.00000	0.00000	0.00000	0.00000
46	19.96176	21.27143	8.91389	6.42029	24.66437	60.60112	5.94503	0.36890	0.08267	0.00057	0.00000	0.00000	0.00000	0.00000	0.00000
47	19.01733	19.12689	9.59104	6.07891	16.30380	58.47180	10.22502	0.55334	0.19463	0.00010	0.00000	0.00000	0.00000	0.00000	0.00000
48	24.39740	16.90147	8.75924	8.39919	14.54689	42.65291	16.84805	0.20132	0.08082	0.00005	0.00001	0.00000	0.00000	0.00000	0.00000
49	23.45922	15.63773	10.15507	10.09720	14.12183	33.17815	22.17797	0.20652	0.09889	0.00002	0.00001	0.00000	0.00000	0.00000	0.00000
50	18.84994	15.01887	11.15008	11.25822	13.77836	32.53704	31.02522	0.44927	0.00289	0.00006	0.00002	0.00000	0.00000	0.00000	0.00000
51	11.59563	15.65604	14.74415	14.53744	15.89463	36.35456	42.89439	2.94066	0.02559	0.01529	0.00257	0.00000	0.00000	0.00000	0.00000
52	13.76412	19.61021	18.33450	17.74636	18.98373	34.67405	40.82064	3.73621	0.08229	0.03780	0.00302	0.00000	0.00000	0.00000	0.00000
53	14.20172	19.61670	22.73994	20.03880	20.54815	27.76782	33.33268	5.15939	0.23872	0.05960	0.00476	0.00000	0.00000	0.00000	0.00000
54	15.27397	16.34459	22.09365	21.28202	20.65050	25.76099	29.21124	8.73173	2.40134	0.83184	0.00226	0.00000	0.00000	0.00000	0.00000
55	13.44895	15.87806	21.78610	22.77735	20.89159	25.41667	29.32463	10.66870	2.62135	0.87427	0.00027	0.00000	0.00000	0.00000	0.00000
56	10.71803	14.33534	20.11027	21.71958	21.66998	25.72663	30.52383	17.24852	1.61771	0.28664	0.00000	0.00000	0.00000	0.00000	0.00000

57	11.20268	15.53998	17.48012	21.34536	22.90763	24.45464	27.20358	22.04578	2.07071	0.17952	0.00000	0.00000	0.00000	0.00000	0.00000
58	10.72267	15.64812	17.46424	22.29310	22.55104	21.80372	21.02869	18.98349	4.77552	0.58622	0.00000	0.00000	0.00000	0.00000	0.00000
59	10.75476	14.35311	17.61410	22.43188	22.62040	21.47975	19.36100	17.19626	6.59565	0.84224	0.00000	0.00000	0.00000	0.00000	0.00000
60	15.84666	8.79907	14.58499	48.71896	35.72359	5.57715	2.45390	0.92136	0.09763	0.01437	0.00000	0.00000	0.00000	0.00000	0.00000
61	13.61626	9.17794	16.17707	45.47285	35.12069	10.58494	4.03654	1.22845	0.13791	0.03812	0.00000	0.00000	0.00000	0.00000	0.00000
62	13.44485	11.83319	14.27235	32.63058	38.98834	24.98693	5.46345	2.03250	0.16370	0.04880	0.00000	0.00000	0.00000	0.00000	0.00000
63	14.63336	12.51690	11.82154	29.40144	38.62310	25.52235	6.33090	4.05120	0.85100	0.34801	0.00000	0.00000	0.00000	0.00000	0.00000
64	11.89408	12.03187	13.22648	24.26238	37.35652	22.35998	9.75656	8.04306	2.75703	1.25568	0.00000	0.00000	0.00000	0.00000	0.00000
65	15.76908	20.84608	14.45260	18.64061	31.63580	34.79565	9.27217	6.50818	1.01464	0.30871	0.00000	0.00000	0.00000	0.00000	0.00000
66	14.71997	24.15763	13.49842	11.45764	24.50859	55.37005	10.42316	2.39113	0.38198	0.13049	0.00000	0.00000	0.00000	0.00000	0.00000
67	16.44841	19.93096	11.56392	7.59696	18.86397	66.36600	10.30492	2.28435	0.51834	0.00333	0.00000	0.00000	0.00000	0.00000	0.00000
68	19.50296	18.68419	11.16397	7.90270	16.98045	59.47732	10.90767	1.68410	0.39007	0.00018	0.00002	0.00000	0.00000	0.00000	0.00000
69	25.46076	16.57983	10.48505	10.14763	15.17254	36.62022	20.60754	0.34088	0.05859	0.00009	0.00001	0.00000	0.00000	0.00000	0.00000
70	26.07964	17.05080	11.24638	11.31289	15.61121	33.18098	23.82778	0.36033	0.01100	0.00014	0.00004	0.00000	0.00000	0.00000	0.00000
71	27.21075	20.44590	15.72572	16.17820	19.46031	33.72812	26.80723	1.49948	0.13052	0.07845	0.01325	0.00000	0.00000	0.00000	0.00000
72	17.84948	22.40101	21.03279	20.68934	22.28175	29.89657	34.55484	4.01954	0.41547	0.30773	0.06929	0.00000	0.00000	0.00000	0.00000
73	17.97409	22.35209	22.35932	22.65667	26.43183	30.49998	33.02691	4.76159	1.63639	2.01288	0.64776	0.00000	0.00000	0.00000	0.00000
74	19.58132	20.95194	21.78750	23.93738	27.43588	30.04497	30.77768	6.87253	3.51902	2.59120	0.59614	0.00000	0.00000	0.00000	0.00000
75	17.08221	15.01266	15.88156	24.36077	24.81461	30.55897	34.17146	15.34876	7.98108	0.87314	0.00148	0.00000	0.00000	0.00000	0.00000
76	16.19441	14.86122	15.57975	23.55334	24.97337	30.44550	34.13046	18.90522	7.13084	0.67378	0.00028	0.00000	0.00000	0.00000	0.00000
77	11.85810	16.25826	17.01200	21.99584	23.69288	26.06252	28.84946	23.68034	2.62056	0.34041	0.00000	0.00000	0.00000	0.00000	0.00000
78	11.67512	16.48922	17.32525	21.70703	23.45015	24.68698	26.41823	23.94761	2.52052	0.18471	0.00000	0.00000	0.00000	0.00000	0.00000
79	11.21326	13.53822	18.45694	23.25828	23.24199	21.17498	21.37034	24.14440	5.40111	0.09250	0.00000	0.00000	0.00000	0.00000	0.00000
80	15.81862	9.13035	16.53873	46.25843	32.37521	5.10069	2.43003	0.81025	0.13858	0.03977	0.00000	0.00000	0.00000	0.00000	0.00000
81	12.58280	8.43565	14.23818	41.40961	35.54664	14.11941	4.32022	2.09175	0.35988	0.03222	0.00000	0.00000	0.00000	0.00000	0.00000
82	7.61213	8.22479	13.53913	33.36784	38.69125	21.80181	8.47156	4.07290	1.67571	0.70312	0.00000	0.00000	0.00000	0.00000	0.00000
83	8.44621	9.34597	11.04791	28.71279	36.42015	23.06270	11.82319	7.38081	2.27694	0.90883	0.00000	0.00000	0.00000	0.00000	0.00000
84	10.61799	14.96825	12.91656	24.36385	31.26161	27.90100	14.50953	6.64668	2.94044	1.52076	0.00000	0.00000	0.00000	0.00000	0.00000
85	13.05665	20.18710	14.15455	20.27314	27.10438	40.27266	15.26546	4.32007	2.67143	1.74299	0.00000	0.00000	0.00000	0.00000	0.00000
86	12.54310	22.82864	16.69859	14.26435	26.72200	62.17730	13.72968	4.13287	0.60530	0.06374	0.00000	0.00000	0.00000	0.00000	0.00000
87	12.79047	20.75772	17.14295	12.95973	26.47141	67.36934	13.72409	3.98129	0.62707	0.00226	0.00007	0.00000	0.00000	0.00000	0.00000

88	20.63336	19.50982	14.77728	13.42535	21.77927	54.10387	13.96149	3.07227	0.60419	0.00041	0.00005	0.00000	0.00000	0.00000	0.00000
89	36.59271	24.02867	16.23051	16.08911	20.74226	37.83612	24.84712	1.67123	0.01215	0.00091	0.00013	0.00000	0.00000	0.00000	0.00000
90	37.86998	25.13304	17.27617	17.21420	21.56139	36.78941	26.94245	1.91829	0.04805	0.03324	0.00774	0.00000	0.00000	0.00000	0.00000
91	32.26248	22.93732	21.44541	21.35519	22.51175	28.43890	26.62574	2.96108	0.39614	0.30062	0.07057	0.00000	0.00000	0.00000	0.00000
92	26.70728	23.39524	23.18773	23.35289	26.06126	29.31011	29.00820	3.60986	1.23877	1.55739	0.50658	0.00000	0.00000	0.00000	0.00000
93	20.31536	23.27853	23.52530	24.71697	29.20499	31.27294	30.55050	5.55126	2.67775	2.61395	0.79166	0.00000	0.00000	0.00000	0.00000
94	17.43414	16.39065	16.43202	25.86901	28.26573	31.66256	32.87067	14.36818	7.90006	2.99970	0.35477	0.00000	0.00000	0.00000	0.00000
95	16.42444	14.95323	14.68826	25.64389	26.66803	31.34772	33.65126	16.58020	8.83903	2.40499	0.00222	0.00000	0.00000	0.00000	0.00000
96	14.96624	14.79380	15.55377	24.01494	25.52442	31.20114	34.61435	21.01758	7.59449	0.44166	0.00112	0.00000	0.00000	0.00000	0.00000
97	11.98580	16.22364	16.85462	22.21140	24.09937	26.13292	28.54309	24.36526	3.31185	0.46633	0.00000	0.00000	0.00000	0.00000	0.00000
98	10.85250	12.69246	17.42342	23.43823	23.34764	21.57818	23.30359	25.80918	5.49919	0.15659	0.00000	0.00000	0.00000	0.00000	0.00000
99	10.73118	10.12371	17.21550	23.76966	23.56302	21.01524	22.65595	27.22693	7.64657	0.13757	0.00000	0.00000	0.00000	0.00000	0.00000
100	20.87302	10.02899	13.43465	47.81553	29.01676	1.86628	0.52568	0.25083	0.07455	0.01654	0.00000	0.00000	0.00000	0.00000	0.00000
101	15.95341	8.98841	15.47580	44.31328	32.56689	6.23715	2.02998	1.08014	0.41015	0.02576	0.00000	0.00000	0.00000	0.00000	0.00000
102	7.89997	7.21401	14.21959	35.70146	38.27548	18.55361	6.91371	3.67745	1.18214	0.39308	0.00000	0.00000	0.00000	0.00000	0.00000
103	6.58636	7.52118	10.25209	31.45171	37.69707	23.78601	10.91939	5.43049	2.23701	0.82433	0.00000	0.00000	0.00000	0.00000	0.00000
104	8.36798	10.12796	10.85049	26.51281	32.05570	26.82699	15.84507	5.55714	4.06229	2.26871	0.00000	0.00000	0.00000	0.00000	0.00000
105	10.67168	16.64996	13.25050	24.54009	29.27948	33.66206	15.94497	4.93610	2.53436	1.42592	0.00000	0.00000	0.00000	0.00000	0.00000
106	12.04589	20.94172	16.58325	20.63566	30.06574	49.54253	16.63764	4.33552	2.05899	1.29733	0.00000	0.00000	0.00000	0.00000	0.00000
107	12.68297	19.89643	18.73250	17.38162	30.35239	65.14855	15.45071	4.97293	0.85828	0.09283	0.00038	0.00000	0.00000	0.00000	0.00000
108	15.00529	20.60729	18.62922	17.17069	29.89612	63.88111	15.48298	4.73600	0.69848	0.00191	0.00011	0.00000	0.00000	0.00000	0.00000
109	38.18605	27.62318	19.11276	18.95993	24.45366	40.13737	24.46595	2.39179	0.03335	0.00128	0.00018	0.00000	0.00000	0.00000	0.00000
110	39.93431	27.90968	18.94119	18.79118	23.41975	37.81697	25.42542	2.38026	0.04367	0.03517	0.01053	0.00000	0.00000	0.00000	0.00000
111	38.74934	25.81108	20.59239	20.05455	22.65361	32.98084	27.55295	2.64752	0.07508	0.07543	0.02304	0.00000	0.00000	0.00000	0.00000
112	32.24828	23.03792	21.99257	21.65756	22.62097	27.87643	27.63248	3.37900	0.59083	0.38205	0.09154	0.00000	0.00000	0.00000	0.00000
113	28.47982	22.26196	22.86346	23.55363	26.59272	28.78574	28.01879	4.59869	1.96403	1.95212	0.61421	0.00000	0.00000	0.00000	0.00000
114	18.16176	18.71529	20.60839	26.94283	32.21184	31.05138	28.82811	13.27461	5.90100	4.03526	0.60607	0.00000	0.00000	0.00000	0.00000
115	15.91922	15.46762	16.44986	26.51786	28.43880	31.52607	32.63436	17.85003	8.62144	1.96102	0.01619	0.00000	0.00000	0.00000	0.00000
116	14.87175	15.10825	16.00799	26.97422	27.65548	31.08526	32.38566	18.10250	11.61290	3.30860	0.00283	0.00000	0.00000	0.00000	0.00000
117	14.39398	15.52580	18.65667	27.13117	27.95338	28.78348	29.01908	19.33144	14.20201	4.31902	0.00360	0.00000	0.00000	0.00000	0.00000
118	12.75937	13.25838	14.02716	23.21523	24.43287	25.92970	27.98538	24.16329	11.26840	1.50279	0.00001	0.00000	0.00000	0.00000	0.00000

119	11.34303	9.15435	13.48948	21.80179	22.09046	20.82181	23.75949	29.15382	11.94552	0.08530	0.00001	0.00000	0.00000	0.00000	0.00000
120	26.06786	21.61304	21.14235	42.99669	29.95502	1.82772	0.45421	0.16320	0.03371	0.01284	0.00000	0.00000	0.00000	0.00000	0.00000
121	14.47983	8.15553	12.68658	39.22986	33.42058	11.73538	3.08815	1.74097	0.70466	0.09878	0.00000	0.00000	0.00000	0.00000	0.00000
122	9.85626	6.53680	12.02186	34.70571	35.40330	20.55792	5.87122	2.83242	1.34657	0.44363	0.00000	0.00000	0.00000	0.00000	0.00000
123	9.58604	7.56816	11.44034	32.26031	32.57124	24.16575	10.08416	3.77203	1.52263	0.57197	0.00000	0.00000	0.00000	0.00000	0.00000
124	11.92255	11.71257	14.13738	28.16754	29.89525	30.32339	13.49760	3.68792	2.34137	1.31171	0.00000	0.00000	0.00000	0.00000	0.00000
125	14.42809	14.50730	18.91090	27.74517	29.36082	36.44984	16.21873	2.44288	1.88109	1.28380	0.00000	0.00000	0.00000	0.00000	0.00000
126	13.39418	19.52336	23.24136	25.13146	33.51269	51.10745	17.52479	2.96296	0.25846	0.05729	0.00022	0.00000	0.00000	0.00000	0.00000
127	13.48581	22.45536	22.61564	22.40605	34.13597	54.85567	14.85117	2.79345	0.23557	0.01298	0.00095	0.00000	0.00000	0.00000	0.00000
128	23.08314	25.13057	17.90513	19.78038	26.66603	40.85879	10.31142	0.94435	0.04620	0.00199	0.00017	0.00000	0.00000	0.00000	0.00000
129	36.57785	27.76163	17.77604	18.43120	23.50068	37.93827	20.14332	1.74381	0.02623	0.00107	0.00014	0.00000	0.00000	0.00000	0.00000
130	37.02256	28.80591	19.08056	18.15864	22.10115	35.28020	24.44548	2.75003	0.04980	0.01589	0.00465	0.00000	0.00000	0.00000	0.00000
131	30.51370	25.20819	21.23482	20.02416	21.25025	27.78426	24.38360	3.18943	0.21168	0.13246	0.04105	0.00000	0.00000	0.00000	0.00000
132	31.02753	24.78320	22.86672	21.53276	22.40922	26.66886	26.74178	5.69074	2.01259	0.32302	0.07763	0.00000	0.00000	0.00000	0.00000
133	21.95689	22.75388	24.17885	24.67947	28.61267	27.85725	27.15925	15.47608	6.28475	0.71551	0.13169	0.00000	0.00000	0.00000	0.00000
134	17.09678	18.22415	19.90924	26.81966	30.34400	29.71433	28.30916	17.40313	7.95572	3.22796	0.06414	0.00000	0.00000	0.00000	0.00000
135	15.36731	16.58513	19.89836	28.02386	30.19680	29.15629	27.60953	18.46755	11.88705	4.53311	0.00470	0.00000	0.00000	0.00000	0.00000
136	15.02145	16.15035	20.61261	27.96804	27.90942	27.88976	27.68076	19.69089	16.46501	4.50061	0.00439	0.00000	0.00000	0.00000	0.00000
137	13.89137	14.28555	16.99184	24.74610	25.31078	25.32392	25.47465	22.30474	18.83849	3.62360	0.00470	0.00038	0.00000	0.00000	0.00000
138	12.25867	10.49368	10.43321	19.11384	20.95130	21.49068	23.86825	27.82796	18.03184	0.92207	0.01108	0.00041	0.00000	0.00000	0.00000
139	12.05017	10.00106	9.62352	19.00064	20.79731	21.36615	24.68229	28.84638	16.42497	0.61015	0.03162	0.00255	0.00000	0.00000	0.00000
140	28.29222	25.04839	23.42993	44.01748	29.86340	1.96398	0.47856	0.14421	0.04704	0.02412	0.00000	0.00000	0.00000	0.00000	0.00000
141	27.07782	23.51355	21.93552	42.70086	29.96251	3.66691	0.64588	0.11954	0.02532	0.01211	0.00000	0.00000	0.00000	0.00000	0.00000
142	14.62280	7.58461	12.22891	36.93502	29.23960	17.42410	3.06431	0.41857	0.04126	0.01459	0.00000	0.00000	0.00000	0.00000	0.00000
143	12.70275	6.78988	12.93695	35.02697	29.49429	20.29514	2.85244	0.42899	0.07523	0.02977	0.00000	0.00000	0.00000	0.00000	0.00000
144	13.72367	8.71359	16.46005	33.61881	29.15546	25.24544	4.38889	0.33859	0.13668	0.07657	0.00000	0.00000	0.00000	0.00000	0.00000
145	15.61070	12.80166	20.91552	30.98553	29.90382	33.37306	12.68287	1.18260	0.14750	0.07988	0.00000	0.00000	0.00000	0.00000	0.00000
146	15.84245	16.32344	22.48374	28.64787	30.55846	38.31100	13.43848	1.07968	0.08980	0.04659	0.00000	0.00000	0.00000	0.00000	0.00000
147	16.36713	20.22588	23.27455	26.21112	30.93748	39.54692	13.92806	1.07539	0.02892	0.00393	0.00019	0.00000	0.00000	0.00000	0.00000
148	19.65202	24.22389	19.91894	21.85338	28.85506	39.31934	9.19977	0.53911	0.03385	0.00374	0.00028	0.00000	0.00000	0.00000	0.00000
149	23.84086	25.68863	18.06081	19.75525	25.35025	37.46428	10.63636	0.90712	0.09352	0.00099	0.00011	0.00000	0.00000	0.00000	0.00000

150	25.95559	27.89996	17.21888	17.65315	21.58652	32.68773	13.69673	2.31469	0.36954	0.00163	0.00026	0.00000	0.00000	0.00000	0.00000
151	26.51713	28.80302	19.41451	17.92723	20.29014	29.46660	20.10176	4.01125	0.81542	0.04610	0.01495	0.00000	0.00000	0.00000	0.00000
152	25.38021	30.24439	21.35165	18.58994	20.58353	27.50510	21.77548	6.59959	2.56604	0.30972	0.09760	0.00000	0.00000	0.00000	0.00000
153	30.23088	39.32070	27.22299	20.20353	20.80439	25.69072	24.60257	16.13984	8.30760	0.26080	0.07860	0.00000	0.00000	0.00000	0.00000
154	27.10115	33.89614	26.90353	22.33759	24.23581	26.36425	25.41689	18.06641	8.50560	0.22613	0.04528	0.00000	0.00000	0.00000	0.00000
155	20.07178	21.17794	25.41626	25.90474	27.97937	26.74326	26.28112	20.10196	9.50767	0.84404	0.01099	0.00000	0.00000	0.00000	0.00000
156	16.82985	18.44360	26.56230	28.75286	25.37079	24.81871	25.02026	23.55201	21.42032	3.71486	0.00277	0.00000	0.00000	0.00000	0.00000
157	15.61501	16.73680	23.78899	26.80184	24.34384	24.00287	23.81858	23.83318	22.85909	3.56287	0.00561	0.00044	0.00000	0.00000	0.00000
158	12.97379	12.49757	14.47869	20.39047	21.42873	21.77603	22.21839	25.66819	21.78024	2.87447	0.36133	0.06153	0.00000	0.00000	0.00000
159	11.61989	9.82283	9.40316	17.33279	19.79443	20.52761	23.60058	28.38200	18.94804	3.08368	0.34152	0.03742	0.00000	0.00000	0.00000
160	30.06015	28.69179	27.78633	46.33606	31.24485	2.98529	0.04450	0.00672	0.00108	0.00026	0.00000	0.00000	0.00000	0.00000	0.00000
161	18.99781	10.88608	14.24511	38.89051	27.25788	10.74949	1.05797	0.05057	0.00328	0.00013	0.00000	0.00000	0.00000	0.00000	0.00000
162	12.81669	7.07300	13.82905	34.03527	27.29092	18.23063	1.64254	0.09872	0.01193	0.00018	0.00000	0.00000	0.00000	0.00000	0.00000
163	11.08387	7.34947	15.97805	32.47254	26.86595	18.86904	2.54961	0.05028	0.00644	0.00050	0.00000	0.00000	0.00000	0.00000	0.00000
164	11.67824	10.14260	21.17800	32.05905	27.54662	24.00026	4.55670	0.07474	0.00831	0.00054	0.00000	0.00000	0.00000	0.00000	0.00000
165	14.53769	13.12101	23.56443	31.81565	28.58792	31.49458	10.64298	0.74122	0.00375	0.00055	0.00000	0.00000	0.00000	0.00000	0.00000
166	16.62288	18.76331	23.13276	27.13696	30.08383	35.95260	12.59598	0.77614	0.00401	0.00121	0.00000	0.00000	0.00000	0.00000	0.00000
167	17.74925	22.22099	22.36651	24.32592	29.71826	36.10038	9.11588	0.30269	0.00876	0.00193	0.00012	0.00000	0.00000	0.00000	0.00000
168	21.90743	24.42101	18.83890	21.08955	25.44455	32.97241	8.45782	0.73820	0.04206	0.00050	0.00006	0.00000	0.00000	0.00000	0.00000
169	23.24255	25.70732	17.66115	19.36719	22.33048	30.98084	9.62047	1.67754	0.41177	0.00094	0.00014	0.00000	0.00000	0.00000	0.00000
170	21.04181	30.19790	18.07625	17.34923	19.27748	28.65050	16.74481	5.48815	2.57581	0.00142	0.00038	0.00000	0.00000	0.00000	0.00000
171	21.56408	31.22441	18.89026	17.47160	19.23573	28.15625	17.89957	6.64535	3.23497	0.03498	0.01131	0.00000	0.00000	0.00000	0.00000
172	29.48546	40.19473	26.02801	19.09311	19.45612	25.40436	22.64946	15.63409	8.15804	0.04574	0.01458	0.00000	0.00000	0.00000	0.00000
173	30.65976	40.62304	27.98511	19.94593	19.92067	24.85307	24.13997	18.11217	9.18041	0.11259	0.02381	0.00000	0.00000	0.00000	0.00000
174	28.94412	36.71767	32.16139	20.39369	19.04828	22.33779	24.18178	22.29518	8.71322	0.15920	0.00793	0.00000	0.00000	0.00000	0.00000
175	21.67645	24.61853	34.05687	24.84281	20.36776	20.53960	23.58773	27.14169	15.96526	1.29100	0.00309	0.00000	0.00000	0.00000	0.00000
176	18.80515	20.77450	29.85777	27.40162	22.14654	22.17288	23.50378	26.84968	21.84834	2.37006	0.02370	0.00216	0.00000	0.00000	0.00000
177	15.87447	16.74183	24.35324	24.89783	21.44259	21.63049	22.21353	26.54866	24.46341	3.43249	0.42069	0.07156	0.00000	0.00000	0.00000
178	10.37423	9.79135	11.19572	15.39653	17.77199	19.23971	21.63822	25.43995	22.25768	10.69747	3.58244	0.18595	0.00000	0.00000	0.00000
179	9.66475	8.89221	9.74987	14.32966	17.20324	18.88961	22.11429	25.28371	20.97430	12.67916	4.45884	0.27053	0.00000	0.00000	0.00000
180	26.89968	20.34271	26.74180	51.89115	25.90470	2.91565	0.02245	0.00924	0.00198	0.00092	0.00000	0.00000	0.00000	0.00000	0.00000

181	26.35104	19.88575	26.57150	50.98977	25.78136	3.59372	0.03453	0.00166	0.00027	0.00012	0.00000	0.00000	0.00000	0.00000	0.00000
182	15.51099	10.10549	17.19063	34.56295	26.56826	14.55508	0.57312	0.00988	0.00131	0.00004	0.00000	0.00000	0.00000	0.00000	0.00000
183	11.58619	8.35659	17.75050	31.85040	26.30317	16.98070	1.84041	0.01884	0.00070	0.00005	0.00000	0.00000	0.00000	0.00000	0.00000
184	9.76767	9.08364	20.75806	31.89184	25.49399	18.88631	2.65519	0.02623	0.00133	0.00033	0.00000	0.00000	0.00000	0.00000	0.00000
185	9.70984	10.25789	23.38384	31.96196	26.29992	21.40918	4.49750	0.05721	0.00167	0.00021	0.00000	0.00000	0.00000	0.00000	0.00000
186	16.33694	14.37023	24.36913	29.31943	26.02456	24.75093	3.09953	0.06020	0.00085	0.00055	0.00000	0.00000	0.00000	0.00000	0.00000
187	20.02852	18.73650	21.92729	24.85093	27.29131	27.29267	4.26708	0.19833	0.00235	0.00090	0.00000	0.00000	0.00000	0.00000	0.00000
188	20.49707	22.59260	20.40013	22.31483	24.79003	28.01914	8.60865	1.28477	0.04533	0.00045	0.00005	0.00000	0.00000	0.00000	0.00000
189	22.07822	23.68404	19.83525	20.20062	21.91402	25.77256	10.81502	3.30912	0.29922	0.00079	0.00012	0.00000	0.00000	0.00000	0.00000
190	20.99449	26.53677	18.72070	19.02778	20.54747	26.65376	12.78801	4.91020	1.91621	0.00105	0.00028	0.00000	0.00000	0.00000	0.00000
191	18.43139	30.59604	17.44135	17.21241	18.87543	27.41642	13.89309	7.29513	3.76907	0.00157	0.00052	0.00000	0.00000	0.00000	0.00000
192	19.48687	31.74314	18.27806	17.31893	18.76982	27.01099	14.77412	8.80116	4.49133	0.00236	0.00051	0.00000	0.00000	0.00000	0.00000
193	32.38689	41.13032	25.17086	18.42280	18.01093	23.15567	23.20750	18.45211	8.76556	0.00287	0.00060	0.00000	0.00000	0.00000	0.00000
194	31.26701	38.11788	29.98711	18.86794	17.51558	20.87300	24.00377	22.29726	7.94856	0.01392	0.00008	0.00000	0.00000	0.00000	0.00000
195	24.79620	28.71348	34.90938	20.75163	17.76214	18.43530	23.53322	26.35180	6.29900	0.25019	0.00009	0.00000	0.00000	0.00000	0.00000
196	21.38208	24.44616	33.63110	23.48747	18.72769	18.69484	22.61657	27.82478	13.42461	1.19638	0.00975	0.00100	0.00000	0.00000	0.00000
197	17.49926	20.41669	27.83309	25.25391	19.69296	19.54763	21.03807	27.56638	22.39474	3.67870	0.54196	0.09204	0.00000	0.00000	0.00000
198	10.04532	11.04985	14.15692	16.07891	16.28553	17.39156	19.64648	22.15212	18.90117	15.41416	8.43401	1.59200	0.00000	0.00000	0.00000
199	9.05184	7.79291	8.93054	13.41824	15.95799	17.93613	20.16365	20.91923	20.46755	21.56289	7.97231	0.77195	0.00000	0.00000	0.00000
200	24.78257	19.81075	29.07514	52.10699	24.77237	3.92168	0.17119	0.00106	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
201	20.99334	13.38945	25.00797	45.21458	24.20290	8.20863	0.08730	0.00028	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
202	15.98487	10.98255	21.19360	34.91313	26.48913	14.66490	0.19676	0.00019	0.00003	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
203	11.69683	9.42343	20.76443	31.75833	25.81088	16.41426	2.06528	0.00020	0.00011	0.00005	0.00000	0.00000	0.00000	0.00000	0.00000
204	9.71511	10.24696	24.43234	31.97970	26.02888	19.66641	3.79582	0.00132	0.00023	0.00014	0.00000	0.00000	0.00000	0.00000	0.00000
205	16.28701	14.67191	26.72329	28.26431	25.30361	22.55860	1.92431	0.01385	0.00024	0.00015	0.00000	0.00000	0.00000	0.00000	0.00000
206	19.39524	16.91840	25.50907	24.91179	25.36882	23.74375	2.12989	0.13846	0.00116	0.00031	0.00000	0.00000	0.00000	0.00000	0.00000
207	19.81542	19.39693	22.35138	22.71873	23.48869	24.34538	5.13721	1.30679	0.01712	0.00033	0.00000	0.00000	0.00000	0.00000	0.00000
208	20.83881	22.25618	20.57692	20.64272	21.81909	24.77338	9.95964	3.28097	0.03714	0.00031	0.00004	0.00000	0.00000	0.00000	0.00000
209	21.58467	23.15631	19.82067	19.96543	21.38881	24.87337	10.62928	3.63890	0.26789	0.00060	0.00015	0.00000	0.00000	0.00000	0.00000
210	19.04473	27.79033	17.64931	17.72712	19.78870	26.37047	12.65308	6.61978	2.80513	0.00080	0.00027	0.00000	0.00000	0.00000	0.00000
211	18.69049	30.70699	17.51206	17.07793	18.64347	26.97855	14.27111	7.92921	3.94662	0.00092	0.00030	0.00000	0.00000	0.00000	0.00000



212	24.98937	33.74823	19.47679	16.43855	16.89632	22.07265	18.57506	14.51098	5.68874	0.00179	0.00041	0.00000	0.00000	0.00000	0.00000
213	30.31896	36.23336	24.37778	16.88195	15.99234	19.36329	24.19169	20.68662	7.14691	0.00391	0.00000	0.00000	0.00000	0.00000	0.00000
214	27.83050	31.41707	30.91094	18.72343	16.46756	17.57803	24.39908	25.06780	5.57769	0.08362	0.00001	0.00000	0.00000	0.00000	0.00000
215	23.15533	26.66537	33.47811	20.34298	17.19850	17.26417	22.44039	26.94216	6.91171	0.73248	0.00642	0.00091	0.00000	0.00000	0.00000
216	16.18329	20.74220	23.49114	20.65808	17.38564	16.06621	19.22726	25.91071	11.29317	2.38476	0.31515	0.05533	0.00000	0.00000	0.00000
217	10.32966	13.62835	15.13470	18.07525	16.74728	15.90843	17.35491	22.68728	16.29260	10.31498	5.79483	1.47956	0.00000	0.00000	0.00000
218	9.06353	7.82187	9.50543	13.88768	15.79836	17.67143	18.56730	18.21769	19.62768	23.38315	10.63344	1.66942	0.00000	0.00000	0.00000
219	8.91185	7.44888	9.28951	13.76781	15.83523	17.80369	18.73189	18.08950	19.71630	23.80541	10.87926	1.70609	0.00000	0.00000	0.00000
220	17.13355	15.20687	32.55582	54.03619	18.95688	3.48869	0.31951	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
221	17.04305	14.18771	31.32841	51.03913	20.48383	5.77944	0.26535	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
222	18.51698	13.15855	25.44719	37.85109	27.56672	12.77612	0.08352	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
223	17.53952	12.05405	25.63192	35.17627	28.20770	13.26281	0.12260	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
224	17.18228	13.10797	27.31139	32.12068	28.73878	14.48006	0.75737	0.00093	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
225	17.64649	15.14482	28.96198	28.85334	27.43598	18.50001	1.06067	0.00390	0.00004	0.00003	0.00000	0.00000	0.00000	0.00000	0.00000
226	15.24131	18.19472	28.75782	21.66790	23.76959	21.72912	1.21078	0.03519	0.00016	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000
227	15.04139	19.44689	27.44643	20.67566	22.68562	22.51880	3.08495	0.68720	0.00080	0.00004	0.00000	0.00000	0.00000	0.00000	0.00000
228	14.06415	21.62749	21.12332	20.15997	19.83557	22.77104	10.28671	2.16611	0.01188	0.00006	0.00000	0.00000	0.00000	0.00000	0.00000
229	13.72891	18.31585	18.39769	19.46636	20.01149	23.37307	11.90458	2.19513	0.01719	0.00005	0.00000	0.00000	0.00000	0.00000	0.00000
230	15.41441	17.22547	16.57051	18.69589	20.19983	24.06230	13.00544	2.78598	0.26028	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
231	17.10416	25.64720	16.66975	16.31934	17.68708	24.25335	14.26810	6.70075	2.17487	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
232	15.90655	24.74375	20.07039	13.42498	12.99421	21.10625	27.79119	10.52627	2.16742	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
233	19.04111	27.21351	19.95823	10.31935	9.20764	13.36443	41.36081	18.10075	3.06144	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000
234	22.70864	30.11236	19.04241	11.03150	10.81793	12.20206	37.43996	21.60237	4.08989	0.02756	0.00000	0.00000	0.00000	0.00000	0.00000
235	22.00230	26.04238	27.23924	18.21367	15.89100	16.24612	23.98764	25.22530	5.95782	1.00699	0.00000	0.00000	0.00000	0.00000	0.00000
236	17.32180	21.24954	22.38557	19.33329	16.41632	15.24476	18.67305	26.17166	9.47318	1.61705	0.00574	0.00080	0.00000	0.00000	0.00000
237	14.47189	18.55025	17.20419	18.11886	15.79977	14.40241	16.73151	25.85682	13.72349	3.76932	0.66001	0.11750	0.00000	0.00000	0.00000
238	9.75582	11.10497	11.68403	16.06071	16.15978	16.00035	16.66462	20.38513	16.94483	14.38801	8.28016	2.10330	0.00000	0.00000	0.00000
239	9.75162	8.05938	9.40469	14.20141	16.17565	17.78295	17.77361	17.13974	19.16712	23.38100	10.53119	1.89057	0.00000	0.00000	0.00000
240	11.19143	13.47048	36.77226	49.65107	17.57748	5.59292	0.37564	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
241	13.84881	14.05425	33.55992	47.21140	21.53694	8.38101	0.35492	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
242	16.92487	13.24304	31.31354	33.41121	26.58474	9.30734	0.09627	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

243	18.13953	13.88732	31.53333	31.54953	27.70104	10.13022	0.11389	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
244	19.44045	15.74617	29.34261	30.01727	28.41890	14.09102	0.49234	0.00108	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
245	15.12782	20.14812	30.45098	23.71579	25.21524	18.48553	1.19358	0.01120	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
246	12.64396	21.70210	30.32244	20.13390	22.28921	20.51663	2.51611	0.25644	0.00037	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
247	11.72107	23.01789	23.27476	18.86863	19.02784	21.83060	8.57434	1.07660	0.00017	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
248	12.34364	20.27323	18.90507	19.45437	18.76720	22.73712	11.45110	1.14537	0.00273	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
249	11.51175	17.38754	17.51234	19.19572	19.21397	23.13303	12.39166	1.11662	0.02234	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
250	17.40679	14.55284	13.35316	15.71250	16.93905	21.78559	19.14180	2.45432	0.14020	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
251	20.72531	16.79914	17.00329	10.78725	10.24330	17.77891	36.21610	8.65301	0.17155	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
252	13.81541	20.32095	18.00563	7.63811	6.71187	12.22661	47.62461	15.39258	0.29801	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
253	14.00574	20.71150	18.19632	7.48299	6.46954	11.33692	48.08701	17.48792	0.50464	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000
254	18.39802	19.74997	13.25626	8.13085	7.94752	10.01398	35.64906	22.53152	1.02842	0.02132	0.00000	0.00000	0.00000	0.00000	0.00000
255	21.81143	18.10048	14.05546	12.21980	10.77270	12.65369	20.31260	25.88575	5.72295	1.12876	0.00000	0.00000	0.00000	0.00000	0.00000
256	16.69525	18.28419	14.91294	16.27498	14.03381	13.12682	15.65222	27.35108	13.40232	2.20856	0.01007	0.00141	0.00000	0.00000	0.00000
257	17.49282	16.83689	13.00304	14.75893	13.48843	12.26776	14.47648	27.07210	16.90350	2.35254	0.30473	0.05344	0.00000	0.00000	0.00000
258	16.92254	13.41363	11.18477	12.81957	13.37974	14.14067	15.45107	21.26268	20.27352	10.76157	5.20224	1.08004	0.00000	0.00000	0.00000
259	15.83418	11.88904	10.43135	12.54245	13.35823	15.24296	16.37592	19.39120	21.01883	14.05706	6.88750	1.33122	0.00000	0.00000	0.00000
260	8.86878	11.94096	38.01699	47.95484	16.21260	5.93586	0.79498	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
261	8.93196	11.91838	37.96063	47.56680	16.33751	6.08422	0.78848	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
262	14.77447	14.25384	34.68604	35.40755	23.20663	7.42063	0.33518	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
263	16.11156	14.00374	32.35909	31.49599	25.04145	7.99183	0.28376	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
264	18.26217	14.66442	31.31629	31.41056	27.06775	9.69059	0.22301	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
265	20.33799	20.06294	28.07625	27.73917	26.37199	14.13191	0.84797	0.00338	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
266	11.34556	28.89214	30.02255	19.78219	21.13498	17.40434	2.65895	0.06964	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
267	12.31537	28.20537	25.56862	19.04148	19.28698	19.56478	6.33593	0.53064	0.00010	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
268	14.11532	23.58398	20.44745	19.39683	18.42550	21.54983	10.01034	1.07125	0.00052	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
269	13.79891	20.04287	18.16466	18.99219	19.21029	20.93087	11.40061	1.03433	0.00220	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
270	18.75882	15.16344	14.11164	16.10995	18.17543	19.38174	17.25692	2.27521	0.00566	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
271	27.44535	16.60990	12.09168	11.07328	12.18913	18.79383	30.93381	6.37058	0.01352	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
272	23.37452	16.49104	15.41426	10.12904	9.73985	17.91318	37.58460	8.43020	0.02040	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
273	17.45584	17.19757	14.47087	7.72598	7.32367	13.71528	43.84949	18.41438	0.02649	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

274	20.37153	16.06267	10.53965	7.99049	8.11923	12.62813	35.38841	21.77729	0.08024	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
275	24.73805	14.58991	10.71660	9.61242	8.90605	12.56252	22.13581	26.03104	2.16494	0.01448	0.00000	0.00000	0.00000	0.00000	0.00000
276	24.29892	13.79268	10.37502	10.26174	9.01882	11.30175	15.74349	30.72170	6.91507	0.38507	0.00000	0.00000	0.00000	0.00000	0.00000
277	24.80065	14.68880	10.32955	9.74662	9.18614	10.27773	14.01459	32.39171	16.70115	0.37201	0.00157	0.00022	0.00000	0.00000	0.00000
278	23.67433	16.22837	11.09358	10.47816	9.93579	11.04316	14.08045	26.59576	21.73309	1.76081	0.15728	0.02735	0.00000	0.00000	0.00000
279	23.28566	16.79425	11.60069	11.03452	10.67111	11.65713	13.64066	23.41092	22.16827	2.94529	0.49700	0.08518	0.00000	0.00000	0.00000
280	8.45299	12.85439	38.34811	42.95020	15.91194	6.50858	0.76069	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
281	8.16945	17.80950	38.03982	41.62035	16.12256	5.64520	0.92802	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
282	12.74280	18.45034	34.69862	34.38934	21.19307	5.76017	0.31839	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
283	15.30908	16.58559	32.65795	32.05187	22.75168	7.15619	0.48313	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
284	24.18148	22.88100	26.18070	23.90739	19.86994	10.26951	0.69387	0.00061	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
285	20.80162	31.23528	26.18159	21.30713	19.64950	11.51152	1.72085	0.00685	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
286	16.65445	35.33938	24.75656	20.56619	19.33553	11.49258	2.22571	0.08202	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
287	16.21206	31.24997	22.01332	19.42664	18.26476	15.83701	5.92652	0.51786	0.00014	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
288	21.75205	23.17401	18.48526	17.37566	17.89134	19.46718	8.34644	0.92623	0.00055	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
289	25.68630	21.17537	17.03595	15.99324	17.73736	18.32435	8.46550	1.23565	0.00078	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
290	27.14911	17.51790	13.30634	12.92049	15.18562	17.29369	24.20200	4.92613	0.00041	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
291	27.65056	16.33383	11.74507	10.62010	11.61932	18.52427	32.00282	7.26309	0.00817	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
292	26.78542	15.21552	10.27826	9.29298	9.57423	17.52364	35.09882	15.77242	0.01549	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
293	26.55106	14.74014	9.83247	9.44503	10.07314	15.90679	32.20218	20.42751	0.04390	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
294	25.64551	14.10813	9.70494	9.58727	9.86351	14.97950	27.64031	23.15883	0.51253	0.00004	0.00000	0.00000	0.00000	0.00000	0.00000
295	27.29889	14.71047	10.22171	9.39721	8.41007	11.33502	16.74319	33.35629	5.96184	0.00310	0.00000	0.00000	0.00000	0.00000	0.00000
296	28.10472	15.05459	9.93102	8.70302	7.94438	10.67067	15.82641	35.32795	11.57658	0.01119	0.00000	0.00000	0.00000	0.00000	0.00000
297	26.89408	15.62988	10.50096	9.41522	8.94495	10.82929	14.17330	31.76416	17.21746	0.09384	0.00134	0.00019	0.00000	0.00000	0.00000
298	25.38191	19.20203	12.42978	10.70151	10.18035	12.09506	13.99517	25.46601	21.46226	2.76457	0.00390	0.00052	0.00000	0.00000	0.00000
299	23.21572	20.84435	13.59347	11.90444	11.13954	12.77516	13.39648	21.50897	22.41366	5.03109	0.06411	0.00850	0.00000	0.00000	0.00000
300	7.34506	14.18828	42.38603	42.23738	16.11802	6.44158	0.93123	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
301	7.68364	18.66934	40.97543	39.45914	16.64932	5.05434	0.60070	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
302	7.98452	24.23161	36.28795	37.24408	16.37570	4.29269	0.22545	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
303	7.74004	25.32949	33.78947	36.27892	14.83729	4.95933	0.90657	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
304	17.53507	22.87198	28.09270	28.01714	15.53166	7.89137	1.21059	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

305	24.03587	25.62009	24.48679	22.32230	17.74652	8.44412	0.72387	0.00041	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
306	20.42459	33.31590	24.06940	19.63877	18.09833	9.38432	1.12159	0.00531	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
307	18.39485	36.69744	22.35988	17.69605	15.79585	9.94923	4.30600	0.03972	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
308	23.64724	29.01364	18.79877	15.65098	14.97128	14.40388	5.47893	0.10038	0.00038	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
309	26.19325	22.69440	17.71063	15.92058	17.28018	17.63583	7.44304	0.91591	0.00087	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
310	26.90278	21.81061	17.20035	15.68078	17.30093	17.80561	8.38288	1.22546	0.00050	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
311	31.07634	21.03275	15.30719	12.61015	12.36946	17.44224	24.61373	5.48909	0.00060	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
312	30.69860	18.59854	13.15397	11.18127	10.82624	16.33368	29.92005	9.71399	0.00812	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
313	27.04848	15.00768	10.11204	9.68829	10.23395	15.88729	31.55917	19.94863	0.02498	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
314	28.10172	14.72808	10.56998	9.37771	9.74938	15.47138	29.04269	21.71078	0.27397	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
315	30.61555	16.45047	11.47476	9.38521	9.43561	13.00278	20.78261	29.64981	3.23705	0.00004	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
316	28.91359	15.46466	10.83534	9.35301	8.93470	12.14585	16.88629	34.65962	7.39057	0.00102	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
317	29.21621	15.92076	10.63242	9.04350	8.51845	11.91272	16.27510	33.75665	11.46000	0.00266	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
318	27.21115	20.12202	12.93668	10.76389	10.43559	13.24337	14.60167	27.28881	19.28559	2.67463	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
319	26.99742	22.48158	14.59590	12.42997	12.29187	14.61429	13.74148	23.36088	19.31463	3.64432	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000
320	7.55655	19.37161	42.99541	37.67059	17.30059	4.98843	0.13285	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
321	7.78094	23.48671	39.95129	37.30532	16.91373	4.12482	0.08645	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
322	7.49282	24.61088	35.02618	36.68508	15.19590	4.46483	0.71556	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
323	7.58656	25.09011	33.24984	35.03767	13.65232	5.48817	1.09663	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
324	16.52520	26.51030	28.57706	25.16857	13.27385	5.87910	0.81961	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
325	20.73521	30.74418	25.57813	22.23238	13.07807	4.58729	0.20719	0.00011	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
326	19.93429	35.10934	21.45972	19.02655	12.59106	5.86801	3.09089	0.00434	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
327	21.06137	34.57801	20.57027	16.15686	13.77328	8.30248	4.04069	0.00919	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
328	25.98941	27.56376	18.41240	14.61178	13.62943	15.75669	6.61336	0.05787	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
329	31.05399	28.42713	21.06794	14.73547	13.23173	17.75041	4.94139	0.06950	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
330	34.82131	31.39883	23.31135	14.73048	12.29558	15.08585	7.06893	0.81104	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
331	37.01423	26.93546	20.80948	15.19977	13.02377	12.41361	15.86915	4.10505	0.00006	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
332	35.92261	24.24930	19.03596	14.45191	12.34707	12.68169	20.00736	7.10492	0.00193	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
333	32.30288	18.03309	13.70863	11.07929	10.82750	14.08168	22.94644	17.17807	0.02628	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
334	30.88052	16.24077	12.15299	9.99477	10.85311	14.59464	21.75031	22.42010	0.91197	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
335	30.96716	16.55345	12.12887	9.55890	9.51453	13.52615	18.23152	30.60710	4.53776	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

336	30.51081	17.11802	11.96101	9.96189	10.11242	13.48673	17.31531	31.79340	5.75839	0.00064	0.00000	0.00000	0.00000	0.00000	0.00000
337	32.42937	21.08744	15.71434	14.59138	16.57088	19.84829	14.44848	23.81911	10.69755	0.01420	0.00000	0.00000	0.00000	0.00000	0.00000
338	31.34941	26.46491	19.02470	17.66741	20.10929	21.81369	12.30560	19.13232	15.42384	2.57594	0.00000	0.00000	0.00000	0.00000	0.00000
339	29.79138	28.03208	19.90358	18.37748	21.31193	19.92954	10.78226	16.70684	14.82424	3.82987	0.00000	0.00000	0.00000	0.00000	0.00000
340	7.28444	18.72794	52.75413	36.82389	17.39879	4.71460	0.15869	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
341	7.69445	19.96965	48.08324	36.31907	17.97750	4.28189	0.05280	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
342	7.93383	23.93780	39.75071	36.17474	14.11045	3.91731	0.07648	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
343	7.33146	24.70718	37.46033	35.02261	12.00070	4.72004	0.72435	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
344	9.31151	27.15506	33.29294	28.23114	10.49257	3.33460	0.95891	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
345	15.33296	32.68720	28.24589	22.71853	9.04488	1.07846	0.09935	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
346	16.07523	33.12299	26.54781	22.53386	9.12969	1.15498	0.34556	0.00006	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
347	20.32294	33.93731	21.83815	19.18798	9.34945	4.01602	3.27880	0.00164	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
348	26.05120	32.51449	20.97716	14.71027	10.82362	6.80544	3.66212	0.00504	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
349	32.12284	33.11557	23.51752	14.40982	10.73775	9.80511	2.03789	0.01284	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
350	35.54696	34.61304	24.49119	13.68784	10.55312	13.85373	4.18625	0.15380	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
351	39.32990	32.71933	23.45433	14.69572	11.18656	13.59449	6.54559	0.65977	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
352	41.40113	30.48056	21.89294	16.17180	13.76103	12.60651	7.49414	1.00507	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
353	41.66213	29.95046	22.14772	16.70860	15.58527	12.48981	7.79227	1.52048	0.00022	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
354	38.01770	22.01515	15.27956	11.26726	14.64658	19.39983	8.93186	7.64877	0.00792	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
355	36.56573	20.29250	13.69230	10.30657	15.18435	21.12860	8.89905	9.98376	0.37148	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
356	31.38772	23.77851	17.18675	17.01727	22.43570	20.36273	8.05263	11.56757	2.15753	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000
357	32.21459	25.20870	19.05471	19.52869	25.33375	23.83966	8.37310	11.66321	4.12874	0.00015	0.00000	0.00000	0.00000	0.00000	0.00000
358	37.53353	31.57728	24.76109	25.57128	32.36173	28.66915	6.38026	5.77075	5.79303	0.02184	0.00000	0.00000	0.00000	0.00000	0.00000
359	45.01159	36.98478	26.69095	25.88306	35.07498	27.33909	5.06543	4.68982	4.86862	0.05190	0.00000	0.00000	0.00000	0.00000	0.00000
360	7.69810	19.92446	52.28979	34.81544	17.52763	5.02178	0.06926	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
361	7.84447	21.37009	49.93922	34.54326	14.74699	4.29155	0.04179	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
362	8.29338	23.57062	44.78669	33.74725	11.68630	3.34038	0.03477	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
363	8.86961	29.16610	51.98731	20.46003	6.44519	1.72040	0.01742	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
364	10.79747	33.28861	46.12341	14.31481	5.12041	0.21204	0.02799	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
365	15.67674	31.02212	26.38890	20.18903	6.94659	0.16547	0.02002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
366	18.54755	28.61824	23.31769	16.36682	5.00520	0.23573	0.11134	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

367	24.51509	29.18530	22.04294	13.58198	6.97552	1.92626	0.18908	0.00055	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
368	28.43337	32.27260	21.35088	15.10594	9.90722	3.94031	0.77115	0.00524	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
369	34.30236	35.68311	22.67626	12.36979	9.07931	7.55486	1.36551	0.00496	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
370	39.73853	33.97974	20.72239	11.42535	10.16202	9.70137	1.75558	0.01613	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
371	42.92064	30.61181	19.85461	13.81541	13.94071	11.13062	3.25224	0.31513	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
372	45.23249	31.00180	21.32250	17.01617	17.50769	9.39760	3.41705	0.40701	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
373	44.81957	28.92860	19.44444	15.97401	19.13221	12.95341	1.88393	0.22659	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
374	41.78172	25.24709	16.86612	13.35695	18.96510	21.03127	2.94430	0.17651	0.00953	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
375	34.85279	27.68357	18.40605	17.41437	25.77583	25.46405	2.34827	0.05272	0.00307	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
376	32.19555	28.91525	21.41946	21.98124	29.58783	19.86316	1.21079	0.01878	0.00453	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
377	34.23451	28.29170	23.78312	26.24400	30.63177	17.33861	2.24141	1.36113	1.48831	0.00001	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
378	41.59005	34.05756	27.46045	28.48106	32.86573	18.30438	4.38905	2.34172	2.11239	0.00011	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
379	45.37590	38.53047	28.25855	26.17643	35.30673	21.41998	4.86430	2.20211	1.54288	0.00319	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
380	7.65143	20.18871	53.28130	34.52121	17.49037	4.80096	0.03304	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
381	7.68590	20.35538	52.62828	34.29390	17.38926	4.81415	0.02128	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
382	7.98131	22.14582	49.65240	33.22854	12.89982	3.99423	0.01494	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
383	8.58894	28.97857	54.33620	20.03391	5.74707	1.77382	0.00351	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
384	10.37781	32.06995	52.93577	11.52477	4.43776	0.73967	0.00417	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
385	13.27405	30.99285	39.19183	12.41505	4.13746	0.05959	0.00400	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
386	18.03467	27.34866	24.56416	13.65927	4.22517	0.03706	0.01668	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
387	19.92635	27.87908	22.73084	13.54528	4.53182	0.30198	0.15485	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
388	25.85627	29.30841	21.64541	12.54901	6.70723	2.14888	0.30198	0.00006	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
389	31.82032	35.06482	20.50738	12.33484	8.48780	2.73062	0.32122	0.00085	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
390	39.02354	33.89411	18.81986	9.50448	9.97536	5.50912	0.49917	0.00800	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
391	41.60977	33.01856	18.34392	10.53490	10.74285	7.48932	1.14355	0.02006	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
392	45.71117	30.76919	19.63887	15.45762	17.44253	8.52549	0.82176	0.00549	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
393	46.81429	30.65235	20.36364	17.13836	19.57878	8.78018	0.69560	0.01102	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
394	45.08239	27.88036	18.53070	15.50229	20.23375	13.80018	1.45277	0.00107	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
395	38.42399	27.28255	18.05865	15.69928	22.33556	22.84950	2.59151	0.00063	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
396	32.96113	28.64983	21.45141	21.29953	28.58239	19.97627	2.11917	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
397	33.41221	28.90134	22.74091	24.64712	29.55396	16.74037	1.20861	0.00088	0.00080	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

398	35.47375	29.39276	25.45502	27.01398	29.98180	11.92816	2.50745	0.44967	0.06024	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
399	42.48224	34.83168	27.63852	28.52043	32.48861	14.18100	3.02452	0.66476	0.17078	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

## INDICE

<b>Sumario</b> .....	3
<b>Capítulo 1 Presentación</b> .....	4
<b>1.1 Introducción</b> .....	4
<b>1.2 Objetivos</b> .....	4
<b>1.3 Antecedentes</b> .....	5
<b>1.4 Alcances</b> .....	7
<b>1.4.1 Relación con otros trabajos desarrollados</b> .....	7
<b>1.4.2 Delimitación misma del trabajo</b> .....	8
<b>1.5 Importancia del trabajo</b> .....	8
<b>1.6 Estructura del trabajo</b> .....	9
<b>Capítulo 2 Definición del problema</b> .....	10
<b>2.1 Definición del objeto de estudio</b> .....	10
<b>2.2 Definición del ámbito de estudio</b> .....	10
<b>2.3 Formulación de pregunta eje</b> .....	11
<b>2.4 Diferencias con los OCR</b> .....	11
<b>Capítulo 3 Marco Teórico</b> .....	13
<b>3.1 Reconocimiento de patrones</b> .....	13
<b>3.1.1 Conceptos básicos referidos al reconocimiento de patrones</b> .....	13
<b>3.1.2 Conceptos de diseño y metodologías</b> .....	14
<b>3.2 Redes Neuronales</b> .....	19
<b>3.2.1 Introducción</b> .....	19
<b>3.2.2 Teoría del Perceptrón</b> .....	20
<b>3.2.3 Teoría del SOFM</b> .....	22
<b>3.3 Esquema general de la implementación desarrollada</b> .....	25
<b>3.3.1 Algoritmo para determinación de solución</b> .....	27
<b>Capítulo 4 Diseño de la Interfase</b> .....	30
<b>4.1 Problemas por resolver para la implementación</b> .....	30
<b>4.1.1 En la escritura de los números</b> .....	30
<b>4.1.2 En la Digitalización de los números</b> .....	33
<b>4.2. Diseño de la Interfase para digitalizar los dígitos</b> .....	36
<b>4.2.1 Algoritmo propuesto</b> .....	36
<b>4.3 Elección del vector de características para entrenar al Perceptron</b> .....	40



4. 3. 1Objetivo.....	40
4. 3. 2Condiciones a cumplir .....	40
4. 3. 3Alternativas.....	41
4. 3. 4Elección Realizada .....	44
4. 4 Elección del vector de características para entrenar al SOFM.....	44
4. 4. 1Objetivo .....	44
4. 4. 2Condiciones a cumplir .....	44
4. 4. 3Alternativas para la extracción del Vector de Características (feature vector) .....	45
<b>Capítulo 5: Estrategia de Solución para el entrenamiento de los modelos .....</b>	<b>49</b>
5. 1 Alcances.....	49
5. 2 Una aproximación inicial haciendo uso del Perceptron .....	49
5. 2. 1El modelo.....	50
5. 2. 2Resultados Obtenidos .....	57
5. 2. 3Conclusiones .....	57
5. 2. 4Limitaciones del Modelo.....	58
5. 3 El modelo mediante el SOFM .....	59
5. 3. 1Antecedentes .....	59
5. 3. 2El modelo.....	62
5. 3. 3Medida de la performance y mecanismos de rechazo.....	75
5. 3. 4Aspectos relevantes en la mejora de solución inicial.....	76
5. 3. 5Conclusiones .....	79
5. 3. 6Limitaciones del modelo .....	80
5. 4 Conclusiones .....	80
<b>Capítulo 6 Consideraciones Finales.....</b>	<b>81</b>
6. 1 Recomendaciones .....	81
6. 2 Diseño de un Clasificador.....	81
<b>ANEXOS</b>	
ANEXO 1 .....	84
ANEXO 2 .....	85
ANEXO 3 .....	89
ANEXO 4 .....	90
ANEXO 5 .....	97
<b>INDICE .....</b>	<b>111</b>
<b>BIBLIOGRAFIA.....</b>	<b>113</b>

## **BIBLIOGRAFIA**

- [1]** Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition", *Neural Computation* 1, 541-551, 1989, Massachusetts Institute of Technology, U.S.A.
  
- [2]** Stefan Knerr, Leon Personazz, and Gerard Dreyfus, "Handwritten Digit Recognition by Neural Networks with Single-Layer Training". *IEEE Transactions on Neural Networks*, vol. 3, No. 6, November 1992.
  
- [3]** Teuvo Kohonen, "The Self -Organizing Map", *Proceedings of the IEEE*, vol. 78, No. 9, September 1990, pp 1464-1480.
  
- [4]** J.T. Tou, R.C. Gonzalez, "Pattern Recognition Principles", Addison Wesley Publishing Company, Inc., Advanced Book Program, Reading, Massachusetts, U.S.A., 1974.
  
- [5]** James.A. Freeman, David M. Skapura, "Neural Networks, Algorithms, Applications, and Programming Techniques", Addison Wesley Publishing, Inc., Reading, Massachusetts, U.S.A., 1991.
  
- [6]** Jian-Kang Wu, "Neural Networks and Simulation Methods", Marcel Dekker Inc., New York, U.S.A., 1995.
  
- [7]** Michael Chester, "Neural Networks a Tutorial", Prentice Hall, New Jersey, U.S.A., 1993.