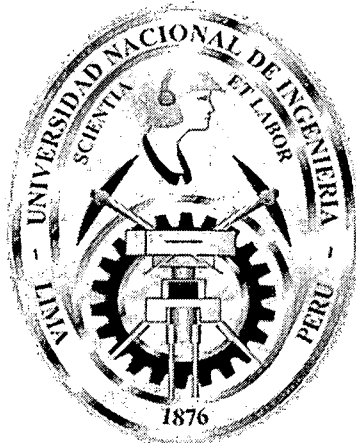


UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y SISTEMAS
SECCIÓN DE POSGRADO



**IDENTIFICACIÓN DE PATRONES DE CONSUMO EN SISTEMAS DE
NEGOCIOS UTILIZANDO REDES NEURONALES**

TESIS

Para optar el Grado Académico de
MAESTRO EN CIENCIAS CON MENCIÓN EN:
INGENIERIA DE SISTEMAS

JOSUÉ ANGULO PÉREZ

LIMA – PERÚ

2011

Digitalizado por:

**Consortio Digital del
Conocimiento MebLatam,
Hemisferio y Dalse**

DEDICATORIA

A mi hija Abigail, con infinita ternura

A María Ulda por su gran apoyo

A mis queridos padres: Leonidas y Eudocia

A mis hermanos: Elizabeth, Eva, Rosa,
Elmer, María, Pedro, Daniel y Eulalia.

AGRADECIMIENTO

Agradezco en forma muy especial a mí asesor el Dr. Pedro C. Espinoza Haro, quien motivó mi interés por las Redes Neuronales, por sus palabras de estímulo en los momentos difíciles que me tocó vivir, por su apoyo y la revisión del presente trabajo.

A mis profesores de Post Grado de la Facultad de Ingeniería Industrial y Sistema de la Universidad Nacional de Ingeniería.

A la Facultad de Ingeniería Industrial y Sistemas por su cálida acogida.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTO	II
ÍNDICE	III
INDICE DE FIGURAS.....	VIII
INDICE DE TABLAS.....	IX
INDICE DE ANEXOS.....	X
DESCRIPTORES TEMÁTICOS.....	XI
RESUMEN EJECUTIVO.....	XIII
INTRODUCCIÓN	XIV
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	1
1.1 Introducción	1
1.2 Identificación del Problema	1
1.3 Definición del Problema	2
1.4 Importancia del Problema	2
1.5 Justificación del Problema	2
1.6 Delimitación del Problema	3
1.7 Objetivos de la Investigación	3
1.7.1 Objetivo General	3
1.7.2 Objetivos Específicos	3
1.8 Formulación de la Hipótesis	4
1.8.1 Hipótesis General	4
1.8.2 Hipótesis Específicas	4
1.8.2.1 Hipótesis Específica N° 1	4
1.8.2.2 Hipótesis Específica N° 2	4

1.8.2.3 Hipótesis Específica N° 3	4
1.8.3 Identificación de las Variables.....	5
1.8.3.1 Hipótesis General	5
1.8.3.1.1 Variable Independiente	5
1.8.3.1.2. Variable Dependiente	5
1.8.3.1.3. Variables Intervinientes	5
1.8.3.2 Hipótesis Específica	5
1.8.3.2.1. Hipótesis Específica N° 1	5
1.8.3.2.1.1 Variable Independiente	5
1.8.3.2.1.2. Variable Dependiente	5
1.8.3.2.1.3. Variables Interviniente	5
1.8.3.2.2. Hipótesis Específica N° 2	5
1.8.3.2.2.1. Variables Independientes	5
1.8.3.2.2.2. Variable Dependiente	5
1.8.3.2.2.3. Variables Interviniente	5
1.8.3.2.3. Hipótesis Específica N° 3	5
1.8.3.2.3.1. Variables Independientes	5
1.8.3.2.3.2. Variable Dependiente	5
1.8.3.2.3.3. Variables Interviniente	5
1.8.4 Operacionalización de las Variables	6
1.8.4.1. Hipótesis General	6
1.8.4.2. Hipótesis Específica.....	6
1.8.4.2.1. Hipótesis Específica 1	6
1.8.4.2.2. Hipótesis Específica 2	6
1.8.4.2.3. Hipótesis Específica 3	7
1.9 Metodología de la investigación.....	7
1.9.1 Método de Investigación	7
1.9.2 Población y Muestra.....	7
CAPÍTULO II: MARCO TEÓRICO	9
2.1 Antecedentes del Problema	9
2.2 Modelo Matemático de la Demanda del Consumidor	10

2.3 Data Mining.....	11
2.3.1 Definición de Data Mining	11
2.3.2 El Proceso de Data Mining	12
2.4 Redes Neuronales	13
2.4.1 Redes Neuronales Biológicas	13
2.4.2 Redes Neuronales Artificiales	15
2.4.3 Modelo Matemático de una Neurona Artificial	16
2.4.3.1 Modelo Matemático para una capa de Neuronas Artificiales.....	17
2.4.3.2 Funciones de Transferencia.....	18
2.4.3.3 Modelo Matemático de Redes Neuronales Multicapas.....	20
2.4.4 Hitos Históricos de la Evolución de las Redes Neuronales...	21
2.4.5 Clasificación de las RNAs.....	22
2.5 La Red Backpropagation.....	24
2.5.1 Introducción	24
2.5.2 La Regla Delta Generalizada.....	25
2.5.3 Funcionamiento del Algoritmo.....	25
2.5.4 Adición de un Momento en la Regla Delta Generalizada...	26
2.5.5 Fases en el Aprendizaje de la Red Neuronal Backpropagation.....	27
2.5.6 Especificaciones de una Red Neuronal de Backpropagation.	27
2.5.7 Estructura y Aprendizaje de la Red Backpropagation.....	29
2.5.8 Consideraciones sobre el Algoritmo de Aprendizaje	33
2.5.9 Control de Convergencia	33
2.5.10 Dimensiones de la Red	34
2.5.11 Aplicaciones de la Redes de Backpropagation.....	34
2.6 Redes de Kohonen	35
2.6.1 Introducción	35
2.6.2 Arquitectura de una Red de Kohonen.....	36
2.6.3 Funcionamiento de una red de Kohonen.....	38
2.6.4 Aprendizaje de una Red de Kohonen.....	39

2.6.5	Aplicaciones de la Redes de Kohonen.....	41
CAPÍTULO III: DESARROLLO DE UN SISTEMA CLASIFICADOR – AGRUPADOR Y MODELO DE SOLUCIÓN		
3.1	Introducción	43
3.2	Esquema del sistema Clasificador - Agrupador.....	45
3.2.1	Entradas del sistema.....	45
3.2.2	Elementos de procesamiento	45
3.2.2.1	Esquema del entrenamiento	46
3.2.2.2	Esquema de Clasificación y Agrupamiento.....	47
3.2.3	Salidas del sistema	48
3.2.3.1	Esquema de consultas y reportes.....	48
3.3	Factores de innovación del sistema clasificador agrupador	49
3.4	Modelo de Solución	49
 CAPÍTULO IV: IDENTIFICACIÓN DE PATRONES DE CONSUMO EN UN SISTEMA DE NEGOCIO UTILIZANDO REDES NEURONALES ARTIFICIALES		
4.1	Fase N°1: Selección de Objetivos.....	52
4.1.1	Presentación del Sistema de Negocios - Morgana Café Lounge	52
4.1.2	Identificación de los Objetivos	56
4.2	Fase N°2: Preparación de los Datos	57
4.2.1	Selección de los datos	57
4.2.2	Pre Proceso de los Datos	58
4.2.3	Transformación de los datos	58
4.3	Fase N°3: Construcción del Modelo	63
 CAPÍTULO V: ANÁLISIS DE LOS RESULTADOS		
5.1	Etapa N° 1: Estudio de Segmentación	65
5.1.1	Aplicación de la Herramienta: NNClust [1993].....	65
5.1.2	Resultados Obtenidos al Final del Análisis.....	71
5.1.3	Conclusiones del uso de la herramienta.....	81

5.2	Etapa N° 2: Estudio del Agrupamiento.....	81
5.2.1	Aplicación de la Herramienta: NNClust	81
5.2.2.	Resultados obtenidos	83
5.2.2.1	Resultados de la Primera Compilación.....	83
5.2.2.2	Resultados de la Segunda Compilación.....	85
5.2.2.3	Resultados de la Tercera Compilación.....	86
5.2.2.4	Análisis de los Resultados.....	88
5.2.2.5	Conclusiones del uso de la Herramienta.....	89
	CONCLUSIONES Y RECOMENDACIONES.....	90
	Conclusiones.....	90
	Recomendaciones.....	91
	Glosario de Términos	93
	REFERENCIAS BIBLIOGRÁFICAS.....	95
	Anexos.....	100

ÍNDICE DE FIGURAS

Figura 2.1 Proceso Data Mining	12
Figura 2.2 Neurona Biológica	14
Figura 2.3 Modelo Matemático de una R.N. artificial	16
Figura 2.4 Modelo Matemático para una capa de Neuronas Artificiales.	17
Figura. 2.5. Modelo Matemático de una Red Multiacpa.....	20
Figura 2.6. Red Backpropagation	29
Figura. 2.7. Arquitectura de una Red de Kohonen	36
Figura 2.8. Función de interacción lateral entre neuronas.....	37
Figura 3.1.Sistema clasificador – agrupador.....	43
Figura 3.2. Cuadro Pictórico de los Procedimientos del Negocio.....	44
Figura 3.3. Esquema del sistema clasificador – agrupador.....	45
Figura 3.4. Esquema de entrenamiento.....	46
Figura 3.5. Esquema de Clasificación y Agrupamiento.....	48
Figura 4.1. Presentación del Sistema de Negocios - Morgana Café Lounge	55
Figura 4.2. Ubicación del local.....	56
Figura 4.3. Modelo Conceptual de la Base de datos de Morgana Café – Lounge.....	59
Figura 4.4. Modelo Lógico de la Base de datos de Morgana Café – Lounge.....	60
Figura 5.1. Interfaz Gráfica de NNClass (Data).....	68
Figura 5.2. Interfaz Gráfica de NNClass (Calc).....	69
Figura 5.3. Interfaz Gráfica de NNClass (Output).....	70
Figura 5.4. Interfaz Gráfica de NNClass (Variable franja).....	71

Figura 5.5 Valores asociados a la variable franja.....	71
Figura 5.6 Resultados del análisis	73
Figura 5.7. Interfaz Gráfica de NNClass (Profile) Variable cantidad...	74
Figura 5.8 Valores de resultados de la variable Cantidad.....	74
Figura 5.9 Representación Gráfica de Cantidad.....	76
Figura 5.10 Cuadro de Análisis de la variable Mozo.....	76
Figura 5.11 Valores de resultados de la variable Mozo.....	77
Figura 5.12 Representación Gráfica de la variable Mozo.....	77
Figura 5.13 Cuadro de Análisis de la variable Cliente.....	79
Figura 5.14 Resultados del análisis de la variable Cliente	79
Figura 5.15 Representación Gráfica de la variable Cliente	81
Figura 5.16 Unidad de Interfaz Gráfica de NNClust.....	82
Figura 5.17 Gráfico de la Primera Compilación.....	83
Figura 5.18 Valores finales de Parámetro de Aprendizaje.....	85
Figura 5.19 Gráfico de la Segunda Compilación del NNClust.....	85
Figura 5.20 Gráfico de la Tercera Compilación del NNClust.....	87
Figura 5.21 Valores finales de Parámetro de Aprendizaje.....	88

INDICE DE TABLAS

Tabla 1.1: Operacionalización de las variables en la Hipótesis General	6
Tabla 1.2: Operacionalización de las variables en la Hipótesis específica Nº 1	6
Tabla 1.3: Operacionalización de las variables en la Hipótesis específica Nº 2	6
Tabla 1.4: Operacionalización de las variables en la Hipótesis específica Nº 3	7
Tabla. 2.2. Hechos significativos de la historia de las redes neuronales.	22
Tabla 4.1. Pre Proceso de los Datos	58
Tabla 4.2. Transformación de datos	62
Tabla 4.3 Transformación de datos	63
Tabla 5.1 Listado de Rubros y Productos.....	66
Tabla 5.2. Listado de Resultados: a, b.....	69
Tabla 5.3. Listado de Resultados de la Variable Franja.....	72
Tabla 5.4. Listado de Resultados de la Variable Cantidad.....	75
Tabla 5.5. Resultados de la Variable Mozo	78
Tabla 5.6. Resultados de la Variable Cliente	80
Tabla 5.7. Resultados de la Primera compilación del NNClust.....	84
Tabla 5.8: Resultados de la Segunda compilación del NNClust.....	86
Tabla 5.9: Resultados de la Tercera compilación del NNClust.....	87
Tabla 5.10: Comparación de Clusters entre Segunda y Tercera Compilación.....	87

INDICE DE ANEXOS

Anexo 1: Requerimiento de Información	101
Anexo 2: Características de las PYMES Industriales	103
Anexo 3: Cuadros Estadísticos	108
Anexo 4: Funcionamiento de las herramientas	109
Anexo 5: Otras Herramientas Disponibles	110

DESCRIPTORES TEMÁTICOS

- Data Mining
- Patrón de Consumo
- Redes Neuronales Artificiales
- Redes de Kohonen
- Backpropagation
- Aprendizaje Supervisado
- Aprendizaje no Supervisado
- Herramientas de Libre Disponibilidad
- Sistema Clasificador-Agrupador

**“IDENTIFICACIÓN DE PATRONES DE CONSUMO EN
SISTEMAS DE NEGOCIOS UTILIZANDO REDES
NEURONALES”**

RESUMEN EJECUTIVO

En la actualidad las empresas generan gran cantidad de información pero desconocen el valor de la misma y no la utilizan en todo su potencial. Dentro de esta información existen patrones y tendencias que pueden indicar necesidades implícitas de los consumidores que pueden representar oportunidades de mejora del negocio.

El propósito de este trabajo es desarrollar un modelo que permita identificar patrones de consumo en un sistema de negocio. Resolveremos este problema de una manera simple y eficiente utilizando redes neuronales artificiales de Kohonen y Backpropagation.

Otro objetivo planteado consiste en aplicar el modelo a un problema real. Específicamente, se trabaja con una base de datos de ventas para identificar patrones implícitos y el comportamiento de las ventas, utilizando herramientas de redes neuronales artificiales de libre disponibilidad, para optimizar la oferta en función de las necesidades de los clientes,

El NNclass, herramienta del Backpropagation, se utiliza para encontrar patrones de consumo de distintos productos en función al día y horario en que los clientes concurren al local. El NNclust, herramienta de aplicación de las redes de Kohonen, se utiliza para analizar la composición de la clientela (estudio de segmentación o "clustering). En base a estos resultados, se propondrán posibles mejoras a la oferta del negocio.

INTRODUCCIÓN

Hoy en día las empresas generan y almacenan gran cantidad de información que en no son aprovechadas, mayormente por desconocimiento o por falta de técnicas para su aprovechamiento y explotación. El problema que se intenta solucionar en el presente trabajo es el deficiente aprovechamiento por parte de las empresas en relación a su base histórica de datos.

Muchas empresas utilizan técnicas de regresión lineal y técnicas estadísticas para buscar patrones de consumo que permitan obtener mayor rentabilidad en los negocios. A diferencia de estas, las redes neuronales, utilizan técnicas de clasificación y segmentación para procesar la información utilizando modelos altamente no lineales basados, esencialmente en técnicas de búsqueda de mínimos locales, que arrojan resultados muy eficientes.

El propósito de este trabajo es desarrollar un modelo clasificador- agrupador que permita identificar patrones de consumo para optimizar la oferta y la demanda y ofrecer un mejor servicio al cliente, lo cual constituye una de las claves en el mundo actual de los negocios.

Para resolver el problema planteado en forma eficiente se utilizarán herramientas basados en redes neuronales y de libre disponibilidad (NNclass y NNclust), las cuales aceleran el proceso de procesamiento de la información. Esto permitirá buscar tendencias que puedan revelar las

necesidades de los consumidores y representar oportunidades de mejora del negocio.

Otra motivación de este trabajo está en mostrar la utilidad y facilidad de la aplicación de las redes neuronales a los negocios, y colaborar en el análisis de datos para pequeñas y medianas empresas que no tiene muchos recursos para utilizar software costoso.

Actualmente, con el avance de la informática, utilizando una computadora con acceso a internet y herramientas de libre disponibilidad, es posible hacer minería de datos, a bajo costo en una PYME, trabajando con bases de datos que no superen 10,000 registros, en caso contrario mediante técnicas de muestreo, puede seleccionarse una muestra representativa de la data.

El presente trabajo pretende contribuir al conocimiento de las redes neuronales, mostrando que es posible aplicarla en PYMES, o en cualquier ámbito comercial, donde sea necesario el análisis de datos. Utilizando la base de datos de una cafetería se demostrará que esta tecnología no es exclusiva de las grandes empresas.

En esa dirección de pensamiento, el trabajo está estructurado en cinco capítulos:

En el Capítulo I se presenta el Planteamiento del problema, identificando, definiendo, justificando y delimitando el problema. También se presentan los objetivos y las hipótesis de la investigación.

En el Capítulo II, se presenta el marco teórico de la investigación considerando temas como Data mining, redes neuronales poniendo énfasis en Backpropagation y las redes neuronales de kohonen.

En el Capítulo III se propone un sistema clasificador – agrupador y el modelo

de solución a seguir para resolver el problema planteado.

En el Capítulo IV, se presenta las tres primeras fases del modelo propuesto para identificar patrones de consumo, a un caso real, utilizando redes neuronales artificiales.

En el Capítulo V, se presenta la cuarta fase del modelo propuesto (Análisis de los resultados), utilizando las herramientas de libre disponibilidad (NNClust y NNClass)

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Introducción

En los últimos años los avances de la tecnología de la información, el soporte computarizado a las operaciones de los negocios y la innovación de los productos hacen que las empresas sean más competitivas. El creciente uso de la informática hace que las empresas almacenen grandes volúmenes de datos que no se aprovechan adecuadamente. Estos datos contienen valiosa información sobre el negocio, que podría ser utilizada para generar mayor rentabilidad y ganancias, lo cual representa la meta de todo empresario. Sin embargo ¿Qué dicen los datos? ¿Cómo se obtienen? , ¿Pueden convertirse en ventaja competitiva para el negocio?, ¿Es posible analizar los datos para identificar patrones de comportamiento de los clientes? ¿Ayudan a la toma de decisiones?

1.2 Identificación del Problema

El uso de la informática hace que las pequeñas y medianas empresas generen y almacenen gran cantidad de datos, con los que se pueden descubrir patrones y tendencias que revelen necesidades de los consumidores que puedan ser convertidas en recursos para la mejora del negocio.

Existen algunas empresas que analizan su información para identificar patrones de negocio utilizando técnicas estadísticas. En este trabajo, se propone un modelo basado en redes neuronales, que requiere una preparación previa de la base de datos y un análisis posterior, pero propone resultados más eficientes.

1.3 Definición del Problema.

El problema del presente trabajo de investigación queda definido de la siguiente forma:

¿Es posible aplicar técnicas de segmentación y clasificación a la base de datos de un negocio para identificar patrones de consumo, que proporcionen oportunidades de mejora, utilizando redes neuronales y herramientas de libre disponibilidad?.

1.4 Importancia del Problema.

La importancia de este trabajo radica en proponer un modelo que permita utilizar herramientas de libre disponibilidad para mostrar la utilidad y la facilidad de aplicación de las redes neuronales a los negocios y colaborar a construir un nuevo paradigma para el análisis de los datos históricos de los negocios, en las PYMES, que cuentan con pequeños capitales y tienen limitaciones para utilizar software muy costosos

1.5 Justificación del Problema.

Los gerentes de las PYMES saben en sus bases de datos existen patrones implícitos que aprovechados eficientemente van a ayudar en la toma de decisiones para enfrentar los nuevos retos.

Las redes neuronales ofrecen un paradigma atractivo para el diseño y

análisis de sistemas adaptativos inteligentes para identificar patrones entre los datos almacenados en las bases de datos que no son aprovechadas. Esto permitirá aplicar el modelo propuesto en el presente trabajo, para hallar patrones de consumo en los datos históricos de las PYMES.

1.6 Delimitación del Problema

En este trabajo se busca proponer un modelo que permita identificar patrones de consumo en sistemas de negocios, del rubro restaurantes y cafeterías, utilizando técnicas de clasificación y agrupamiento de redes neuronales, de Kohonen y Backpropagation, y herramientas de libre disponibilidad.

1.7 Objetivos de la Investigación

1.7.1 Objetivo General

Identificar patrones de consumo en un sistema de negocio utilizando técnicas de clasificación y agrupamiento mediante redes neuronales, y herramientas de libre disponibilidad.

1.7.2 Objetivos Específicos

- 1) Utilizar redes neuronales de Kohonen y Backpropagation para identificar patrones de consumo en un sistema de negocio.

- 2) Utilizar técnicas de clasificación y agrupamiento de redes neuronales para identificar los patrones de consumo en un sistema de negocio.

- 3) Utilizar herramientas de libre disponibilidad, NNclust y NNclass, basadas en tecnologías de las redes neuronales para identificar patrones de consumo orientados al cliente y al producto en un sistema de negocio.

1.8 Formulación de la Hipótesis

1.8.1 Hipótesis General

Mediante la aplicación de técnicas de clasificación y agrupamiento, basadas en redes neuronales, y herramientas de libre disponibilidad a la base de datos de un negocio se logrará identificar patrones de consumo que presenten oportunidades de mejora.

1.8.2. Hipótesis Específicas

1.8.2.1. Hipótesis Específica N°1: Las redes neuronales de Kohonen y Backpropagation permiten identificar patrones de consumo en un sistema de negocio.

1.8.2.2. Hipótesis Específica N°2: Las técnicas de clasificación y agrupamiento de redes neuronales aplicados a la base de datos de un negocio permiten identificar los patrones de consumo.

1.8.2.3. Hipótesis Específica N°3: Las herramientas de libre disponibilidad NNclass y NNclust basadas en tecnologías de las redes neuronales aplicadas a la base de datos del negocio permite identificar los patrones de consumo orientados al producto y al cliente en un sistema de negocios.

1.8.3. Identificación de las Variables

1.8.3.1. Hipótesis General

1.8.3.1.1. Variable Independiente: Base histórica de datos

1.8.3.1.2. Variable Dependiente: Clasificación y agrupamiento, patrones de consumo.

1.8.3.1.3. Variables Intervinientes: Redes neuronales, herramientas de libre disponibilidad.

1.8.3.2. Hipótesis Específica

1.8.3.2.1. Hipótesis Específica N°1

1.8.3.2.1.1 Variable Independiente: Base histórica de datos

1.8.3.2.1.2. Variable Dependiente: Patrones de consumo

1.8.3.2.1.3. Variables Interviniente: Redes neuronales

1.8.3.1.2. Hipótesis Específica N°2

1.8.3.2.2.1. Variables Independientes: Base histórica de datos

1.8.3.2.2.2. Variable Dependiente: Patrones de consumo

1.8.3.2.2.3. Variables Interviniente: Redes neuronales,
Técnicas de clasificación y agrupamiento,

Hipótesis Específica N°3

1.8.3.2.3.1. Variables Independientes: Base histórica de datos

1.8.3.2.3.2. Variable Dependiente: Patrones de consumo

1.8.3.2.3.3 Variables Interviniente: Redes neuronales,
herramientas de libre disponibilidad.

1.8.4. Operacionalización de las Variables

1.8.4.1. Hipótesis General

Variables	Dimensión	Indicadores
VI: Base Histórica de datos de consumo de Clientes.	Tamaño N° de tablas	Economía Eficiencia Veracidad
VD: Patrones de consumo.	Número y tipo de patrones Clasificación y Clusterización	Eficiencia Veracidad Método Utilizado Validez

Tabla N° 1.1: Operacionalización de las variables en la Hipótesis General

1.8.4.2. Hipótesis Específica

1.8.4.2.1. Hipótesis Específica N°1

Variable	Dimensión	Indicadores
VI: Base Histórica de datos de consumo de Clientes.	Tamaño N° de tablas	- Eficiencia - Veracidad
VD: Patrones de consumo	Nro. y tipo de patrones Clasificación y Clusterización	- Eficiencia - Veracidad - Validez - Ajuste

Tabla N° 1.2: Operacionalización de las variables en la Hipótesis específica N° 1

1.8.4.2.2. Hipótesis Específica N°2

Variable	Dimensión	Indicadores
VI: Base Histórica de datos de consumo de Clientes.	Tamaño N° de tablas	- Eficiencia - Veracidad
VD: Patrones de consumo	Nro. y tipo de patrones Clasificación y Clusterización	- Eficiencia - veracidad

Tabla N° 1.3: Operacionalización de las variables en la Hipótesis específica N° 2

1.8.4.2.3. Hipótesis Específica N°3

Variable	Dimensión	Indicadores
VI: Base Histórica de datos de consumo de Clientes.	Tamaño N° de tablas	- Eficiencia - Veracidad - Fácil aprendizaje del software
VD: Patrones de consumo de clientes usando técnicas estadísticas.	Número y tipo de patrones, clasificación y claustrización	

Tabla N° 1.4: Operacionalización de las variables en la Hipótesis específica N° 3

1.9. Metodología de la investigación

1.9.1. Método de Investigación: Se realizará una investigación teniendo en cuenta técnicas y herramientas de las Redes Neuronales.

1.9.2 Población y Muestra: Se tomará como población una base de datos del consumo de los cliente y una muestra de 1000 datos.

- **Técnicas e Instrumentos:** Redes Neuronales y herramientas de libre disponibilidad: Trial Power Designer, Sql Server 2008 R2 Versión Express, Microsoft excel, NNclust y NNclass.
- **Análisis y Tratamiento de los datos:** El análisis de los datos se hace utilizando Trial Power Designed Rational que nos permite construir el modelo conceptual y modelo lógico de la base de datos. A continuación utilizamos SQL Server 2008 R2 versión Express para construir el modelo físico de la base de datos y exportar los datos a Excel obteniendo tablas que permitan seleccionar los datos para iniciar el entrenamiento en las macros. Utilizando NNclust (herramienta del aplicación del SOM, orientada al cliente) y NNclass

(herramienta basada en Backpropagation, orientada al producto) se identificaran patrones de consumo orientados al cliente y al consumo, respectivamente.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes del Problema

De acuerdo a la bibliografía consultada, los enlaces que se mencionan poseen publicaciones referidas a temas generales sobre las redes neuronales:

- <http://www.neurocolt.com/abs/2002/2002.html> [Neurocolt, 2002]
- <http://www.cs.man.ac.uk/~jls/Publications/node4.html> [Shapiro, 2004]
- <http://www.calsci.com/Applications.html> Pattern%20Recog%20Applications [Brain Maker, 2006]

Sobre aplicaciones a los negocios, existen trabajos interesantes en las siguientes publicaciones: [Saha, A. 1998], [Sánchez Comperas E.N., Alania García Alma Yolanda, 2006], [Torres Zorrilla Jorge, 2003] y [García Vega, 2002], que contienen resúmenes de publicaciones sobre redes neuronales aplicadas a los negocios.

En otras publicaciones sobre aplicaciones de las redes neuronales. [Li, 1994] y [Britos, 2005] escriben sobre la factibilidad de aplicar sistemas inteligentes a negocios y proponen algunas, entre las cuales se encuentran aplicaciones a negocios. [Shachmurove y Witkowska, 2000] utilizan redes neuronales para predecir la evolución de la bolsa de valores. [Puppio, 2002], y proponen

aplicar redes neuronales a la tasación inmobiliaria.

Sebastián Gómez de Lima en su Tesis de Grado “Redes Neuronales: Análisis de un caso”, ITBA, 2006, propone aplicar Redes Neuronales para identificar patrones de negocio en una dulcería.

A nivel nacional, entre las bibliotecas consultadas existen pocas publicaciones sobre el tema. En el caso de las aplicaciones se tienen los trabajos de Tesis de Pos Grado presentados por: Vilcapoma Escurra, Edgar S. “Identificación de Patrones de evasión en el sistema de Administración Tributaria, usando tecnología Data Mining”, UNI 2003; Gómez Rodríguez, Gladys G. “Sistemas de Soporte a las decisiones con Data Mining. Caso: Gestión de Deuda en las administraciones tributarias”, U. de Lima, 2007; Renzo José Figueroa Palomino “Modelaje y Análisis comparativo de la situación productiva y competitiva del sector industrial manufacturero mediante la aplicación de Redes Neuronales y Estadística Multivariada”. UNI, 2007; Sandivar Rosas Juana, en su Tesis “Pronóstico de la demanda de postulantes de un Centro Superior Universitario, basado en redes neuronales”, UNI, 2008. Esto demuestra en nuestro país son muy escasos los trabajos publicados sobre el tema.

2.2. Modelo Matemático de la Demanda del Consumidor

La demanda individual del consumidor se puede expresar mediante la siguiente función:

$$D = f(P, I, B_c, B_s, G, C, E, S, \dots) \quad (2.1)$$

donde:

D: Demanda

I : Ingreso

B_c: Bien complementario

B_s: Bien sustituto

G: Gustos y preferencias

C: Clima

E : Edad

S : Sexo

Esta función define al consumo o demanda como un conjunto de factores que se relacionan y dan como producto el grado de aceptación de la adquisición del producto. Los datos pueden ser obtenidos mediante encuestas.

2.3. Data Mining

El Data Mining (Minería de Datos), es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

El Instituto SAS define Data Mining como el “proceso de seleccionar, explorar, modificar, modelar y valorar grandes cantidades de datos con el objetivo de descubrir patrones desconocidos que puedan ser utilizados como ventaja comparativa respecto a los competidores.”

2.3.1. Aplicaciones de Data Mining

El proceso de Data Mining puede aplicarse a una amplia variedad de industrias y proporciona distintas metodologías de análisis según el tipo de problema que queramos analizar. Algunas aplicaciones son:

- **Predicción automática de tendencias y comportamientos.** Data Mining automatiza el proceso de obtención de información predictiva

utilizando bases de datos. Un ejemplo típico de problema predictivo es el marketing dirigido. Data Mining utiliza bases de datos procedentes de campañas promocionales pasadas para identificar los objetivos que con mayor probabilidad harán rentables campañas futuras.

- **Identificación automática de patrones**

Las herramientas de Data Mining permiten analizar base de datos para identificar patrones de ventas de productos aparentemente no relacionados que a menudo se adquieren juntos.

2.3.2. Proceso de Data Mining

El término proceso implica que Data Mining es un conjunto de fases. El proceso consta de las siguientes cuatro fases:

- Selección de los objetivos
- Preparación de los datos
- Construcción del modelo
- Análisis de los resultados

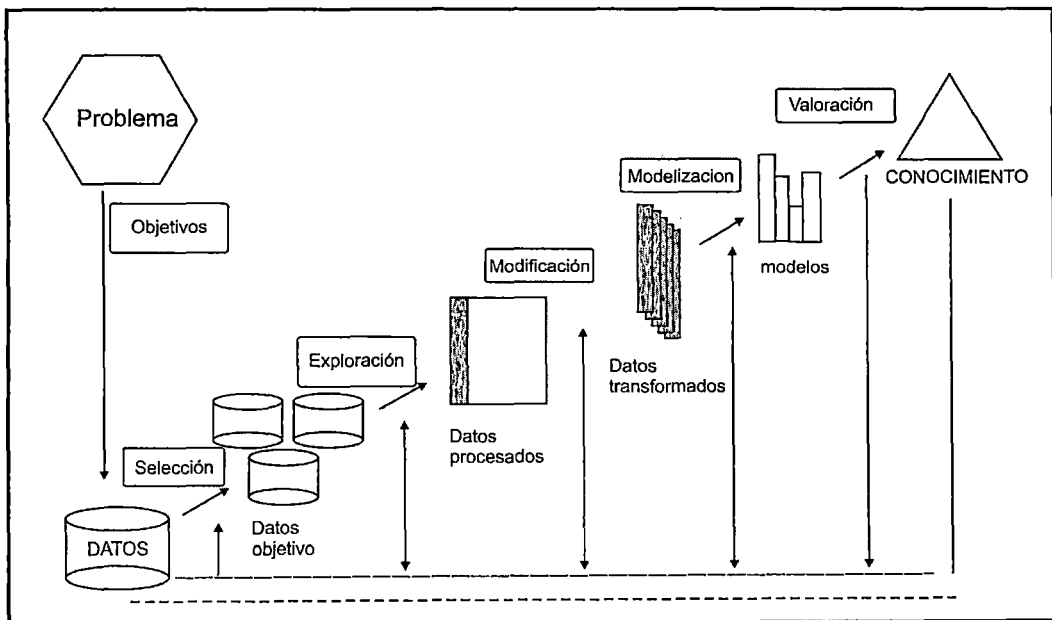


Figura 2.1 Proceso Data Mining

2.4. Redes Neuronales

2.4.1. Redes Neuronales Biológicas

La neurona es el componente básico del sistema nervioso. Tiene tres partes principales: las dendritas, el cuerpo de la célula o soma y el axón. Las dendritas son elementos receptores; el soma transforma esas señales y el axón es una fibra larga que lleva la señal desde el cuerpo de la célula hacia otras.

La sinapsis es el **punto** de contacto entre el axón de una célula y la dendrita de otra. La membrana de la neurona genera impulsos eléctricos y transfiere información a otras neuronas por medio de la sinapsis; la información viaja a lo largo de los axones en breves impulsos eléctricos. El estímulo es creado por la información que llega a través de las dendritas. El impulso se propaga a lo largo del axón por el cambio de conductancia de la membrana.

Las neuronas se comunican cuando el impulso llega al botón sináptico generado la liberación de neurotransmisores por parte de la célula presináptica y la absorción de éstos por la célula postsináptica, que se da en la hendidura sináptica. La sinapsis convierte una señal eléctrica de la neurona presináptica en un proceso químico, que después se reconvierte en una señal eléctrica en la neurona postsináptica.

Se estima que el cerebro humano puede contener 10^{11} neuronas y 10^{15} interconexiones. Las redes neuronales biológicas están constituidas por un gran número de neuronas, conectadas en forma masiva que pueden establecerse como grupos de neuronas activas especializadas en tareas. En la figura 2.2 se puede ver la representación de las neuronas biológicas.

Las últimas teorías sobre el aprendizaje afirman que su efecto en el cerebro resulta de la modificación de las conexiones o sinapsis entre neuronas, lo

que unido al procesamiento de información de la neurona determina el mecanismo básico de la memoria.

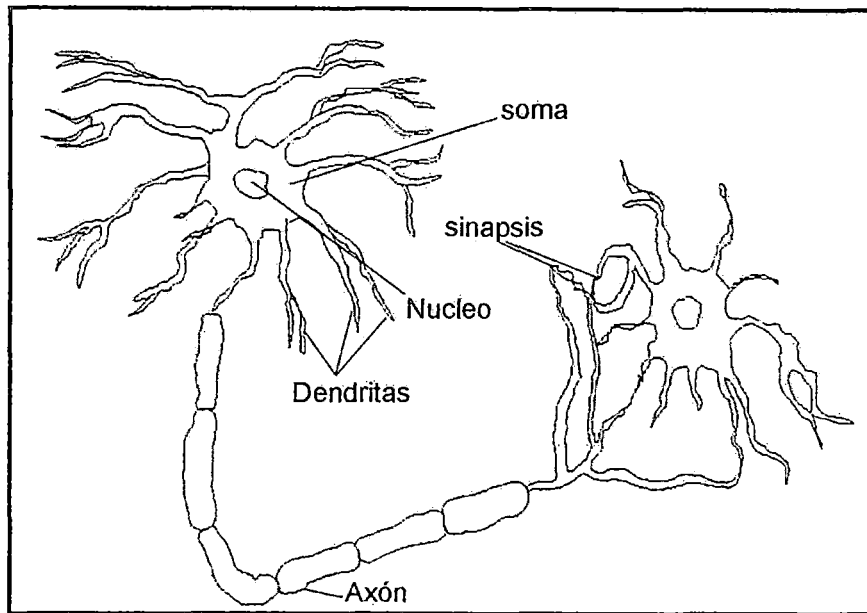


Figura 2.2 Neurona Biológica

Este mecanismo de modificación de los pesos sinápticos es producido en el tiempo por medio de la experiencia del sujeto y se logra variando (aumentando o disminuyendo) el número de neurotransmisores liberados en algunas sinapsis ante un mismo estímulo. De este modo, una neurona cuyo potencial no alcanzaba el umbral ahora sí lo hace y se activa, y viceversa, otra que antes se activaba ahora puede que no lo haga y así la respuesta del sujeto ante una misma situación real es diferente, Se dice entonces que hemos aprendido. Normalmente el aprendizaje requiere además de un maestro que indique cómo realizar bien una tarea. De esta manera, tras un proceso de ensayo – error el sujeto es capaz de realizar correctamente una tarea que antes no podía hacer.

2.4.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) son modelos que intentan reproducir el comportamiento del cerebro, del mismo modo que aquel, realiza una simplificación, averiguando cuáles son los elementos relevantes del sistema. Una elección adecuada de sus características, más una estructura conveniente, es el procedimiento convencional utilizado para construir redes capaces de realizar una determinada tarea. Este modelo posee dispositivos elementales de proceso, las neuronas, al igual que en el modelo biológico.

La propiedad del cerebro humano más utilizada en las RNA es la capacidad de aprender de la experiencia y generalizar a partir de ella.

Bajo esta descripción, las RNA tratan de simular este proceso biológico en un equipo computacional en base al siguiente modelo:

- La información es procesada en elementos básicos llamados neuronas.
- Las señales o inputs, $x_i = 1, 2, \dots, n$ son transmitidas entre las neuronas a través de conexiones.
- A cada conexión se le asocia un peso w que representa la intensidad de la señal recibida correspondiente al input i en la neurona j . La mayor o menor intensidad de cada input proporción a la influencia del mismo en el resultado final.
- Cada neurona j recibe las señales x_1, \dots, x_n mediante sus n dendritas un peso a cada una de ellas w_{1j}, \dots, w_{nj} y se realiza el siguiente proceso

$\sum w_{ij} x_i + b_j$, donde b_j es el “sesgo” o “ganancia” de la neurona j . Este valor es transformado mediante una función denominada de transferencia o activación f_j que corresponde a cada neurona j

- El aprendizaje se realiza mediante el ajuste de los pesos que ponderan las conexiones entre las neuronas que componen la red. El algoritmo de aprendizaje es un método mediante el cual se obtienen los ajustes óptimos evitando tener que supervisar todo el proceso. Este método consta de una serie de pasos iterativos, que en el fondo está basado en la búsqueda de mínimos locales mediante el método del gradiente (o de descenso rápido).
- Para lograr procesar la información, las neuronas se organizan en capas: la capa de entrada transmite las variables input, la capa de salida presenta el output final y entre ellas pueden existir una o más capas denominadas capas ocultas.

2.4.3 Modelo matemático de una neurona artificial

Un modelo matemático para una neurona artificial recoge las características esenciales señaladas anteriormente como se muestra en la figura 2.3 .

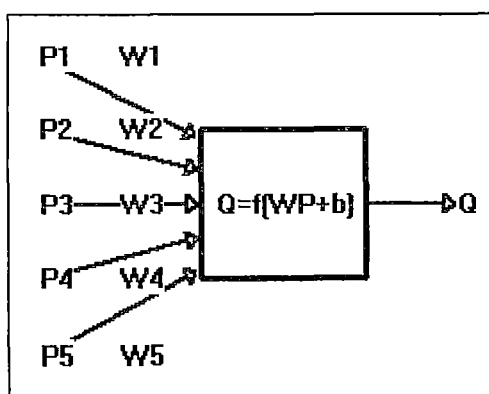


Figura 2.3 Modelo matemático de una red neuronal artificial.

El vector $P=[p_1, p_2, \dots, p_n]$ representa un conjunto de señales, cada una de las cuales es recibida por cada una de las n **dendritas** de una célula. La intensidad con que ingresa una señal p_k está dada por un número w_k denominada **peso**. El conjunto de todos los pesos correspondientes al vector P que llega a la neurona, esta dado por el vector de pesos $W=[w_1, w_2, \dots, w_n]$. El proceso que sigue, dentro del cuerpo de la célula, se representa mediante la suma de los productos de cada señal por su peso, más un valor b , denominado: **ganancia o sesgo de la neurona**. Todo esto se representa en la siguiente expresión:

$$WP + b = w_1p_1 + w_2p_2 + \dots + w_np_n + b$$

Este valor es transformado mediante una función de transferencia $f(s)$ de la neurona, dando lugar a la respuesta de la neurona $Q = f(WP + b)$.

A continuación se presentan los modelos matemáticos para una neurona artificial, una capa de neuronas artificiales y una red neuronal multicapa [Espinoza Haro , Pedro C, 2010].

2.4.3.1 Modelo matemático para una capa de neuronas artificiales

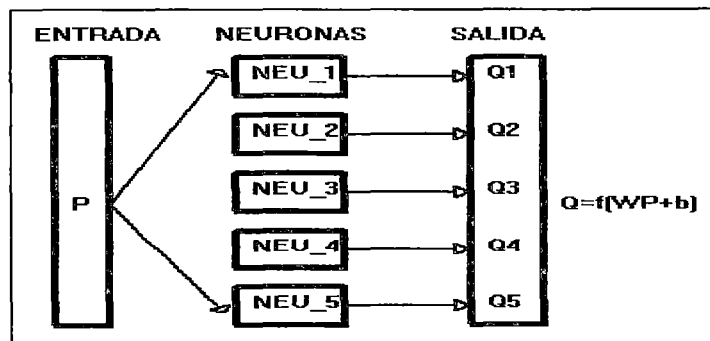


Fig. 2.4 Modelo Matemático para una capa de neuronas.

Para un vector de señales $P=[p_1, p_2, \dots, p_n]$ que llega a cada una de las neuronas de una capa de m neuronas, los pesos asociados se representan mediante una matriz, denominada **matriz de pesos**: $W=[w_{j,k}]$, de m filas por n columnas. La intensidad con que ingresa una señal p_k a cada neurona j , está dada por un número $w_{j,k}$ denominado peso. El conjunto de todos los pesos correspondientes al vector P que llegan a la neurona j ,

esta dado por la fila j : $W_j = [w_{j,1}, w_{j,2}, \dots, w_{j,n}]$ de la matriz de pesos W . Una parte del proceso, dentro de la neurona j , se representa mediante la suma de los productos de cada señal por su peso, más un valor b_j , denominado: **ganancia de la neurona j** . Todo esto se representa por la expresión: $W_j P + b_j = w_{j,1}p_1 + w_{j,2}p_2 + \dots + w_{j,n}p_n + b_j$. Este valor es transformado por la función de transferencia $f_j(s)$ de la neurona j , dando lugar a la respuesta: $Q_j = f_j(W_j P + b_j)$. Esto ocurre en cada neurona. El proceso en la capa de neuronas, se representa mediante el producto de la matriz de pesos W por el vector P , más el vector b de **ganancias de la capa de neuronas**. Cada componente de este vector es transformado por la función de transferencia de cada neurona y que se puede representar mediante una **función vectorial** de valores vectoriales o una transformación $F: \mathbb{R}^m \rightarrow \mathbb{R}^m$, que para cada vector $S = (s_1, s_2, \dots, s_m)$, está definida como :

$$F(S) = [f_1(s_1), f_2(s_2), \dots, f_m(s_m)].$$

En consecuencia la respuesta de la capa será el vector $Q = F(WP + b)$, donde $F(WP + b) = [f_1(W_1P + b_1), f_2(W_2P + b_2), \dots, f_m(W_mP + b_m)]$.

La señal procesada y transformada $Q = F(WP + b)$ es el vector salida de la capa de neuronas. Ver figura 2.4.

El resultado final de una capa de m neuronas, no es otra cosa que una transformación $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$, que transforma una señal $P \in \mathbb{R}^n$ en un vector $Q \in \mathbb{R}^m$, definido por $T(P) = F(WP + b)$

2.4.3.2 Funciones de transferencia

Existen muchas funciones de transferencia empleadas en el diseño de redes neuronales, las más comunes son:

a) Limitador fuerte (Hardlim):

$$a = f(n) = \begin{cases} 1, & \text{si } n \geq 0 \\ 0, & \text{si } n < 0 \end{cases}$$

b) Limitador fuerte simétrico (Hardlims):

$$a = f(n) = \begin{cases} 1, & \text{si } n \geq 0 \\ -1, & \text{si } n < 0 \end{cases}$$

c) Función de transferencia lineal positiva (poslin)

$$a = f(n) = \begin{cases} 0, & \text{si } n < 0 \\ n, & \text{si } n \geq 0 \end{cases}$$

d) Función de transferencia lineal (purelin):

$$a = f(n) = n$$

e) Función de transferencia lineal saturada (satlin)

$$a = f(n) = \begin{cases} 0, & \text{si } n < 0 \\ n, & \text{si } 0 \leq n \leq 1 \\ 1, & \text{si } 1 < n \end{cases}$$

f) Función de transferencia lineal saturada Simétrico (satlins)

$$a = f(n) = \begin{cases} -1, & \text{si } n < -1 \\ n, & \text{si } -1 \leq n \leq 1 \\ 1, & \text{si } 1 < n \end{cases}$$

g) Función de transferencia sigmoideal (logsig):

$$a = f(n) = \frac{1}{1 + e^{-n}}$$

h) Función de transferencia tangente sigmoideal hiperbólica (tansig):

$$a = f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

i) Función de transferencia competitiva (compet)

$$a = f(n) = \begin{cases} 1, & \text{neurona con } n \text{ maximo} \\ 0, & \text{en los demas casos} \end{cases}$$

2.4.3.3 Modelo matemático de Redes Neuronales Multicapas

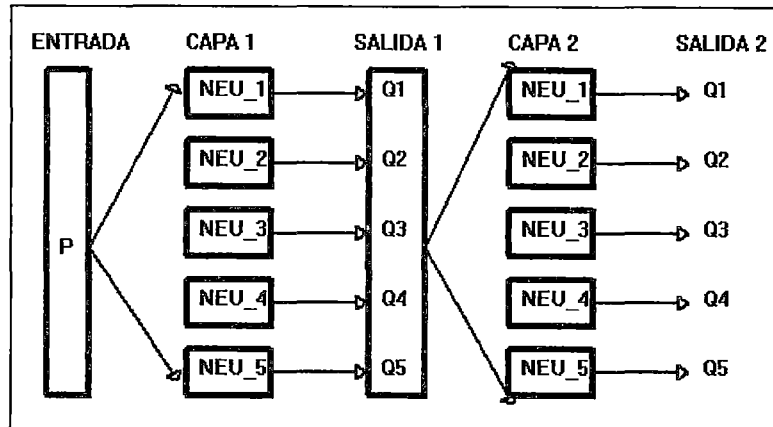


Fig. 2.5 Modelo Matemático de una red multicapa.

Dentro de una red neuronal, los elementos de procesamiento se encuentran agrupados por **capas**. Ver figura 2.5. Una capa es una colección de neuronas. Según la ubicación en la RNA, las capas reciben diferentes nombres.

Capa de entrada: Es la capa formada por el conjunto de neuronas que recibe la señal exterior, esto es el vector de entrada $P = [p_1, p_2, \dots, p_n]$.

Capas ocultas: Estas son las capas que **no reciben** la señal $P = [p_1, p_2, \dots, p_n]$ y tampoco las que emiten las señales finales.

Capa de salida: Es la última capa conectada a capas anteriores y envía la señal al exterior.

Formulación matemática de una red neuronal multicapa.

Supongamos que la RNA tiene N capas. Entonces:

a) Para la **primera capa** se tienen una matriz de pesos $W^1 = [w_{j,k}^1]$ de orden $m_1 \times n$, un vector de ganancias $b^1 = [b_k^1]$ y una función vectorial de funciones de transferencias

$$F^1(S) = [f_1^1(s_1), f_2^1(s_2), \dots, f_{m_1}^1(s_{m_1})].$$

$$\text{La salida es } Q^1 = F^1(W^1 P + b^1)$$

a) Para la **segunda capa** se tendrá una matriz de pesos $W^2 = [w_{j,k}^2]$ de orden $m_2 \times m_1$, un vector de ganancias $b^2 = [b_k^2]$ y una función vectorial de funciones de transferencias

$$F^2(S) = [f_1^2(s_1), f_2^2(s_2), \dots, f_{m_2}^2(s_{m_2})].$$

$$\text{La señal emitida será } Q^2 = F^2(W^2 Q^1 + b^2)$$

b) Para la **N -ésima capa** se tendrá una matriz de pesos $W^N = [w_{j,k}^N]$ de orden $m_N \times m_{(N-1)}$, un vector de ganancias $b^N = [b_k^N]$ y una función vectorial de funciones de transferencias

$$F^N(S) = [f_1^N(s_1), f_2^N(s_2), \dots, f_{m_N}^N(s_{m_N})].$$

c) La señal emitida será

$$Q^N = F^N(W^N Q^{m(N-1)} + b^N).$$

De esta manera se observa que la salida final de la red neuronal de N capas será:

$$P \rightarrow Q^1 = F^1(W^1 P + b^1) \rightarrow Q^2 = F^2(W^2 Q^1 + b^2) \rightarrow \dots \rightarrow Q^N = F^N(W^N Q^{m(N-1)} + b^N)$$

Que es la composición de N transformaciones o N funciones vectoriales no lineales de valores vectoriales. En síntesis es la composición de N transformaciones, que se concatenan o componen:

$$\mathcal{R}^n \xrightarrow{T_1} \mathcal{R}^{m_1} \xrightarrow{T_2} \mathcal{R}^{m_2} \xrightarrow{\dots} \xrightarrow{T_N} \mathcal{R}^{m_N}$$

2.4.4. Hitos Históricos de la evolución de las redes Neuronales

En la tabla 2.1 se muestran los hitos históricos en la evolución de las redes neuronales.

1911	Ramón y Cajal da las primeras ideas de una Red Neuronal
1943	McCulloch y Pitts proponen un modelo neuronal.
1949	Hebb publica el libro "La Organización del Comportamiento", en el cual propone la regla Hebbiana de aprendizaje.
1958	Roseblatt propone el modelo de red de una sola capa que hoy llamamos "Perceptron"
1969	Minsky y Papera muestran las limitaciones del perceptron de una sola capa, postergando esta disciplina.
1982	Hopfield publica una serie de trabajos sobre las redes de Hopfield
1982	Kohonen desarrolla los mapas auto-organizados que actualmente llevan su nombre.
1983	Barto propone el aprendizaje por reforzamiento para las redes neuronales artificiales
1986	Werbos propone el aprendizaje por retro propagación.
1986	El algoritmo de aprendizaje para perceptrones de múltiples capas "Backpropagation" es redescubierto y la disciplina vuelve a surgir.
1990s	Se desarrollan las "Redes de funciones de base radial".
1994	Aparece la teoría del control neuronal con redes recurrentes
2000s	La potencia de los conjuntos de redes neuronales y los "Support Vector Machines" se vuelve notable.

Tabla. 2.1. Hechos significativos de la historia de las redes neuronales.

2.4.5 Clasificación de las RNAs

Los modelos de RNAs pueden ser clasificados atendiendo a diversos criterios.

Según las reglas básicas de entrenamiento-aprendizaje utilizadas, se pueden clasificar en tres tipos fundamentales: supervisado, no supervisado y reforzado.

- **Aprendizaje supervisado:** En estas redes los datos para el entrenamiento están constituidos por un grupo de pares de patrones entrada-salida deseada. Existe un maestro (algoritmo) encargado de verificar que la salida de la red coincida con la respuesta deseada, si no es así se deben modificar los pesos de las conexiones para intentar que la salida obtenida se aproxime lo más posible a la respuesta deseada. Este tipo de redes han sido las más desarrolladas para clasificación de patrones. El Backpropagation utiliza, este tipo de aprendizaje.
- **Aprendizaje no supervisado:** En estas redes se suministran los inputs de entrada, pero no reciben ninguna información que indiquen cuál es la salida. La red modifica los pesos para poder asociar a la misma unidad de salida los vectores de entrada más similares. Estas redes son capaces de auto organizarse. Normalmente, se habla de dos tipos de entrenamiento no supervisado: el hebbiano y el competitivo/cooperativo. El primero es generalmente orientado a detectar las similitudes de las entradas y el segundo más orientado hacia la clusterización. El tipo de red más difundido son las Redes Auto-Organizadas, llamadas SOM (procedente del término inglés Self- Organizing Map), o de Kohonen [Kohonen, 1984].
- **Aprendizaje Reforzado:** En este caso, el supervisor se limita a indicar si la salida ofrecida por la red es correcta o incorrecta, pero no se le indica la cuantía de su error ni la respuesta que debería dar. El tipo de red más extendida de este tipo es la Red de Hopfield.

Según el tipo de variables que usa la red, se pueden clasificar en dos grupos: binarias y continuas.

2.5 La red Backpropagation

2.5.1 Introducción

En 1986, Rumelhart, Hinton y Williams propusieron un modelo de red neuronal que tenía la capacidad de aprender la asociación que existe entre los patrones de entrada a la misma y las clases correspondientes utilizando más niveles de neuronas de los que empleó Rosenblatt en su Perceptrón. Este método conocido como Backpropagation está basado en la generalización de la regla delta.

El algoritmo de propagación hacia atrás o retropropagación es una regla de aprendizaje que se puede aplicar en modelos de redes con más de dos capas de neuronas. Una característica de este algoritmo es la representación interna del conocimiento que es capaz de organizar en las capas ocultas para lograr cualquier correspondencia entre entradas y salidas.

El funcionamiento de una red Backpropagation consiste en un aprendizaje de un conjunto predefinido de pares entrada-salida dados como ejemplo, empleando un ciclo propagación adaptación de dos fases: primero se aplica un patrón de entrada en la capa de entrada de la red que se propaga hacia las capas superiores hasta generar una salida, se compara el resultado obtenido en cada neurona de salida con el valor deseado para esa neurona y se obtiene un error para dicha unidad. A continuación, estos errores se transmiten hacia atrás hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de la neurona intermedia en la salida original. Este proceso se repite capa a capa hasta llegar a la entrada y hasta que cada neurona haya recibido un error que describa su aporte al error total. Según el valor del error recibido, los pesos de las conexiones se reajustan entre cada par de neuronas en la red, de manera de que el error total cometido para ese patrón disminuya.

2. 5.2 La regla delta generalizada

La regla delta propuesta por Widrow en 1960 (regla delta) ha sido extendida a redes con capas intermedias (regla delta generalizada) con conexiones hacia delante (feedforward) y cuyas células tienen funciones de activación continuas, no decrecientes y derivables (lineales o sigmoidales), dando lugar al algoritmo de retro propagación (Backpropagation).

Este algoritmo utiliza una función o superficie de error asociada a la red, buscando el estado estable de mínimo error a través del camino descendente de la superficie de error. Para ello, realimenta el error del sistema para realizar la modificación de los pesos en un valor proporcional al gradiente decreciente de dicha función de error.

2. 5.3 Funcionamiento del Algoritmo

Dada una neurona U_i y su salida y_i , el cambio que se produce en el peso de la conexión que une la salida de dicha neurona con la unidad $U_j(w_{ij})$ para un patrón de aprendizaje p es:

$$\Delta w_{ij}(t+1) = \alpha \delta_{pj} y_{pi} \quad (2.2)$$

Donde el sub índice p se refiere al patrón de aprendizaje concreto y α es la constante o tasa de aprendizaje.

El punto en el que difiere la regla delta generalizada de la regla delta es el valor concreto de δ_{pj} . Por otro lado, en las redes multinivel con capas ocultas no se conoce la salida deseada de las unidades de estas capas para poder calcular los pesos en función del error cometido. Sin embargo, inicialmente sí podemos conocer la salida deseada de las neuronas de salida. Para la unidad de salida U_j , definimos:

$$\delta_{pj} = (d_{pj} - y_{pj}) f'(net_j) \quad (2.3)$$

donde d_{pj} es la salida deseada para la neurona j y el patrón p y net_j es la entrada neta de la neurona j .

Si U_j no es de salida, se tiene:

$$\delta_{pj} = \left(\sum_k d_{pk} w_{kj} \right) f'(net_j) \quad (2.4)$$

donde el rango de k cubre todas las neuronas las que está conectada la salida de U_j . El error que se produce en una neurona oculta es la suma de todos los errores cometidos por las neuronas a las que está conectada su salida multiplicada por el peso de la conexión correspondiente.

2.5.4 Adición de un momento en la Regla Delta Generalizada

En el algoritmo de retro propagación del error utiliza una amplitud de paso que es dada por la tasa de aprendizaje. A mayor tasa, mayor es la modificación de los pesos en cada iteración con lo que el aprendizaje será más rápido pero, por otro lado, puede dar lugar a oscilaciones. Para filtrar estas oscilaciones, se añade en la expresión del incremento de los pesos, un término (momento) β , de manera de que dicha expresión quede:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \delta_{pj} y_{pi} + \beta (w_{ij}(t) - w_{ij}(t-1)) =$$

$$\dot{w}_{ij}(t+1) = \alpha \delta_{pj} y_{pi} + \beta \Delta w_{ij}(t) \quad (2.5)$$

Donde β es una constante que determina el efecto en $t+1$ del cambio de los pesos en el instante t .

Con este momento se asegura la convergencia de la red en un menor número de iteraciones, pues si en t el incremento de un peso era positivo y en $t+1$ también, entonces el descenso por la superficie de error en $t+1$ es

mayor. Sin embargo, si en t el incremento era positivo y en $t+1$ es negativo, el paso que se da en $t+1$ es más pequeño, lo cual es adecuado ya que esto significa que se ha pasado por un mínimo y que los pesos deben ser menores para poder alcanzado.

2.5.5 Fases en el Aprendizaje de la Red Neuronal Backpropagation

Existen tres fases en la aplicación del Backpropagation.

- **Fase de Entrenamiento.** Esta fase consiste en hacer que la red sea capaz de extraer, a partir de ejemplos reales, reglas generales que permitan en el futuro responder adecuadamente a patrones que no han sido vistos anteriormente.
- **Fase de Validación.** Una vez entrenada la red, se realiza la fase de validación, que consiste en correr la red con otro conjunto de datos para comprobar el nivel de entrenamiento alcanzado y ver si responde a los objetivos propuestos.
- **Fase de Test o Prueba.** Es la fase en la cual se ingresa a la red un patrón de entradas con el fin de obtener una clasificación y agrupamiento de los datos. Se evalúa así, una vez comparada la respuesta real y la estimada.

2.5.6. Especificaciones de una Red Neuronal del Backpropagation

En una Red Neuronal Backpropagation las neuronas están agrupadas por capas de modo que todas las neuronas de una capa se conectan con las de la siguiente capa y así sucesivamente, pues las neuronas de una misma capa no tienen ninguna conexión.

Un vector de entrada $x_p = (x_{p1}, x_{p2}, x_{p3}, \dots, x_{pN})^t$ se aplica a las neuronas de la capa de entrada de la red, y ésta distribuirá dichos valores a las neuronas de la capa oculta. Las entradas netas hacia estas neuronas se calculan mediante la siguiente fórmula:

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h \cdot x_{pi} + \theta_j^h \quad (2.6)$$

donde:

- El superíndice h se refiere a magnitudes de la capa oculta;
- El subíndice p se refiere al p-ésimo vector de entrenamiento.
- El subíndice j se refiere a la j-ésima neurona oculta.
- El término θ representa el sesgo que puede considerarse como una entrada más cuyo término asociado es la unidad.
- w_{ji}^h es la ponderación de la señal procedente de la neurona de la capa de entrada i que llega a la i-ésima neurona de la capa oculta h.

net_{pj}^h es la suma lineal de los inputs recibidos ponderados y N representa el número de inputs considerado. El resultado se hace pasar a continuación a través de una función de transferencia, generalmente sigmoideal.

$$i_{pj} = f_j^h (net_{pj}^h) \quad (2.7)$$

Para las neuronas de la capa de salida, la entrada neta y su correspondiente salida se calculan mediante las siguientes expresiones;

$$net_{pk}^o = \sum_{j=1}^L w_{kj}^o \cdot i_{pj} + \theta_k^o \quad (2.8)$$

Donde L el número de neuronas en la capa oculta. Este resultado es de nuevo transformado en la neurona de la capa de salida cuyo resultado es el output definitivo.

$$y_{pk} = f_k^o(\text{net}_{pk}^o) \quad (2.9)$$

donde el superíndice o se refiere a magnitudes de la capa de salida

2.5.7 Estructura y Aprendizaje de la red Backpropagation

En una red Backpropagation [Figura 2.6] existe una capa de entrada con n neuronas y una capa de salida con m neuronas y al menos una capa oculta de neuronas internas. Cada neurona de una capa (salvo la de entrada) recibe entrada de todas las neuronas de la capa previa y genera salida hacia todas las neuronas de la capa siguiente (salvo las neuronas de salida). No existen conexiones hacia atrás feedback ni laterales o autorrecurrentes.

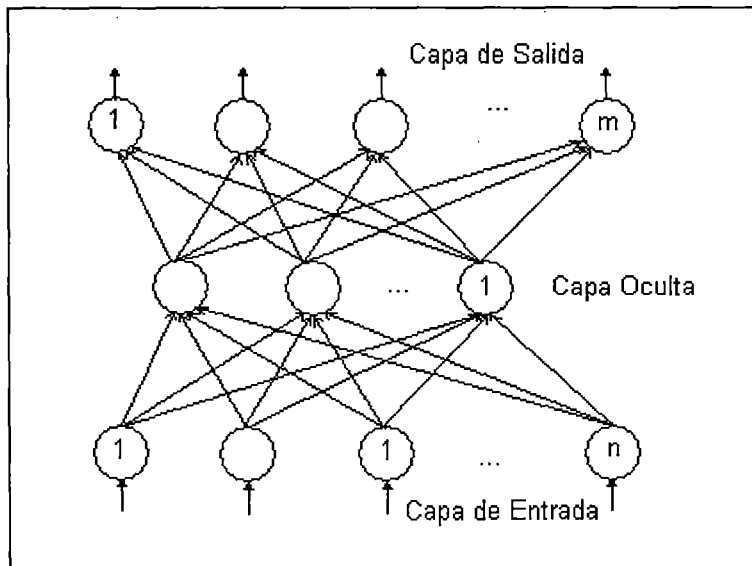


Fig. 2.6 Red Backpropagation

A continuación se presentan los pasos y fórmulas a utilizar para aplicar el algoritmo de entrenamiento .

Paso 1: Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2: Presentar un patrón de entrada: $X_p = x_{p1}, x_{p2}, \dots, x_{pn}$, y especificar las salida deseada que debe generar la red: d_1, d_2, \dots, d_M (si la red se utiliza como un clasificador, todas las salidas deseadas serán 0, salvo una, que será la de la clase a la que pertenece el patrón de entrada).

Paso 3: Calcular la salida actual de la red, para ello presentamos las entrada a la red y se va calculando la salida que presenta cada capa hasta llegar a la capa de salida ésta será la salida de la red y_1, y_2, \dots, y_M . Los pasos a seguir son los siguientes:

- Se calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.
- Para una neurona j oculta:

$$\text{net}_{pj}^h = \sum_{i=1}^N w_{ij}^h x_{pi} + \Theta_j^h \quad (2.10)$$

Donde el índice h se refiere a magnitudes de la capa oculta (hidden), el subíndice p , al p -ésimo vector de entrenamiento, y j a la j -ésima neurona oculta.

- Se calculan las salidas de las neuronas ocultas:

$$y_{pj} = f_j^h (\text{net}_{pj}^h) \quad (2.11)$$

- Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida (capa o: output)

$$\text{net}_{pk}^o = \sum_{j=1}^L w_{jk}^o y_{pj} + \Theta_k^o \quad (2.12)$$

$$y_{pk} = f_k^o (\text{net}_{pk}^o) \quad (2.13)$$

Paso 4: Calcular los términos de error para todas las neuronas.

Si la neurona k está en una capa de salida, el valor de delta es:

$$\delta_{pk}^0 = (d_{pk} - y_{pk}) f_k^0 (\text{net}_{pk}^0) \quad (2.14)$$

La función f debe ser derivable por lo cual no se puede utilizar una función escalón. Existen dos funciones que pueden servir: la función lineal ($f_k (\text{net}_{jk}) = \text{net}_{jk}$) función sigmoidal:

$$f_k = (\text{net}_{jk}) = \frac{1}{1 + e^{-\text{net}_{jk}}} \quad (2.15)$$

La selección de la función de salida depende de la forma de representar los datos de salida: si se requiere que las neuronas de salida sean binarias, se utiliza la función sigmoidal, en otro caso es tan aplicable una función como la otra.

Para la función lineal tenemos $f_k^0 = 1$, mientras que la derivada de una función sigmoidal es:

$$f_k^0 = f_k^0 (1 - f_k^0) = y_{pk} (1 - y_{pk}) \quad (2.16)$$

por lo que el término de error para las neuronas de salida queda:

$$\delta_{pk}^0 = d_{pk} - y_{pk} \quad (2.17)$$

para la salida lineal, y;

$$\delta_{pk}^0 = (d_{pk} - y_{pk}) y_{pk} (1 - y_{pk}) \quad (2.18)$$

Para la salida sigmoidal.

Si la neurona j no es de salida se tiene:

$$\delta_{pj}^0 = f_j^h (\text{net}_{pj}^h) \sum_k \delta_{pk}^0 w_{jk}^0 \quad (2.19)$$

Donde observamos que el error en las capas ocultas depende de todos los

términos de error de la capa de salida. De aquí surge el término de propagación hacia atrás. Para la función sigmoïdal:

$$\delta_{pj}^0 = x_{pi} (1 - x_{pi}) \sum_k \delta_{pk}^0 w_{jk}^0 \quad (2.20)$$

Donde k se refiere a todas las neuronas de la capa superior a la de la neurona j. Así el error que se produce en una neurona oculta es proporcional a la suma de los errores conocidos que se producen en las neuronas a las que está conectada la salida de ésta, multiplicados por el peso de la conexión.

Paso 5: Actualización de los pesos: Utilizando el algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada, ajustando los pesos de la forma siguiente:

Para la capa de salida:

$$w_{jk}^0(t+1) = w_{jk}^0(t) + \Delta w_{jk}^0(t+1); \quad (2.21)$$

$$\Delta w_{jk}^0(t+1) = \alpha \delta_{pk}^0 y_{pj} \quad (2.22)$$

y para los pesos de las neuronas de la capa oculta:

$$w_{ij}^h(t+1) = w_{ij}^h(t) + \Delta w_{ij}^h(t+1); \quad (2.23)$$

$$\Delta w_{ij}^h(t+1) = \alpha \delta_{pj}^h x_{pi} \quad (2.24)$$

En ambos casos, para acelerar el proceso de aprendizaje, se puede añadir un término momento de valor $\beta(w_{jk}^0(t) - w_{jk}^0(t-1))$ en el caso de la neurona de salida, y $\beta(w_{ij}^h(t) - w_{ij}^h(t-1))$ en el caso de una neurona oculta.

Paso 6: El proceso se repite hasta que el término de error

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (2.25)$$

Sea lo suficientemente pequeño para cada uno de los patrones aprendidos.

4.5.8 Consideraciones sobre el Algoritmo de Aprendizaje.

El algoritmo Backpropagation encuentra un valor mínimo de error (local o global) mediante la aplicación de gradiente descendente. Cada punto de la superficie de la función de error corresponde a un conjunto de valores de los pesos de la red. Con el gradiente descendente, siempre que se realiza un cambio en todos los pesos de la red, se asegura el descenso por la superficie del error hasta encontrar el valle más cercano, lo que puede hacer que el proceso de aprendizaje se detenga en un mínimo local de error.

Este algoritmo de entrenamiento busca minimizar la función de error pudiendo caer en un mínimo local o algún punto estacionario sin llegar a encontrar el mínimo global. Sin embargo se debe tener en cuenta que no siempre es necesario encontrar el mínimo global, sino que puede ser suficiente con un error menor al máximo preestablecido.

2.5.9 Control de la Convergencia

Cuando se utilizan las técnicas de gradiente descendente es conveniente avanzar por la superficie de error con incrementos pequeños de los pesos. Esto se debe a que una información de la superficie es local y no se sabe lo lejos o cerca que se está del mínimo. Con incrementos grandes se corre el riesgo de pasar por encima del punto mínimo sin conseguir estacionarse en él. Con incrementos pequeños. Aunque se tarde más, se evita la situación anterior.

La velocidad de convergencia se controla con un parámetro que normalmente debe ser un número pequeño (de 0,05 a 0,25) para asegurar la con-

vergencia y evitar oscilaciones.

El incremento de la velocidad de convergencia se logra añadiendo un término momento consistente en sumar una fracción del cambio anterior a la variación de peso actual.

Si una red deja de aprender antes de llegar a una solución aceptable, se realiza un cambio en el número de neuronas ocultas o en los parámetros de aprendizaje o, simplemente, se vuelve a empezar con un conjunto distinto de pesos originales y se suele resolver el problema.

2.5.10 Dimensionamiento de la red

No existen reglas concretas para determinar el número de neuronas o de capas de una red para resolver un problema concreto. Lo mismo ocurre con el número de patrones de entrenamiento.

En general tres capas son suficientes, sin embargo algunas veces la red aprende con más de una capa oculta. El número de neuronas ocultas que debe tener cada capa es difícil de determinar, no existe regla que lo haga. Si la red se simula por software, se recomienda mantener el número de neuronas por capa oculta lo más bajo que se pueda, ya que cada neurona implica un costo de procesamiento extra.

2.5.11 Aplicaciones de las Redes Backpropagation.

- Identificación de patrones de consumo de distintos productos en función al día y horario en que los clientes concurren al local. En este trabajo se utiliza el NNclass, basado en la red Backpropagation para descubrir patrones de consumo.
- Reconocimiento óptico de caracteres (OCR).

Reconocimiento de patrones, de caracteres de un texto, se puede utilizar una red Backpropagation para aprender las diferentes formas de los caracteres que se representarán como imágenes compuestas por cierto número de píxels, generando la red como salida el código (por ejemplo ASCII) correspondiente al carácter de entrada. El objetivo es que la red reconozca formas de caracteres no utilizadas en la fase de aprendizaje siempre que no difieran demasiado de los patrones mostrados durante el entrenamiento.

2.6 Redes de Kohonen

2.6.1. Introducción

Para identificar patrones de consumo, orientado al cliente, en un sistema de negocio se ha considerado utilizar redes de Kohonen, porque este modelo permite crear mapas topológicos bidimensionales, de modo que, ante los datos de entrada con características comunes, se activen neuronas situadas en zonas próximas a las capas de salida. Esto permite establecer relaciones previamente desconocidas entre conjuntos de datos.

Según las últimas investigaciones, en el cerebro hay neuronas que se organizan en zonas, de tal manera que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales. Por ejemplo, en el sistema visual se han detectado mapas del espacio visual en zonas del córtex (capa externa del cerebro). Esto sugiere, que el cerebro podría tener la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior.

Utilizando estas ideas, T. Kohonen propuso en 1982 un modelo de red neuronal que tenía la capacidad de formar mapas de características de manera similar a como ocurre en el cerebro. El objetivo de Kohonen era demostrar que un estímulo externo por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas.

Este modelo tiene dos variantes, denominadas LVQ (Learning Vector Quantization) y TPM (Topology Preserving Map) o SOM (Self Organizing Map). Ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de éstos, siendo el LVQ de una dimensión y bidimensional e incluso tridimensional la red TPM.

2.6.2. Arquitectura de una Red de Kohonen

La Red de Kohonen tiene dos capas con N neuronas de entrada y M de salida (Figura 2.7)

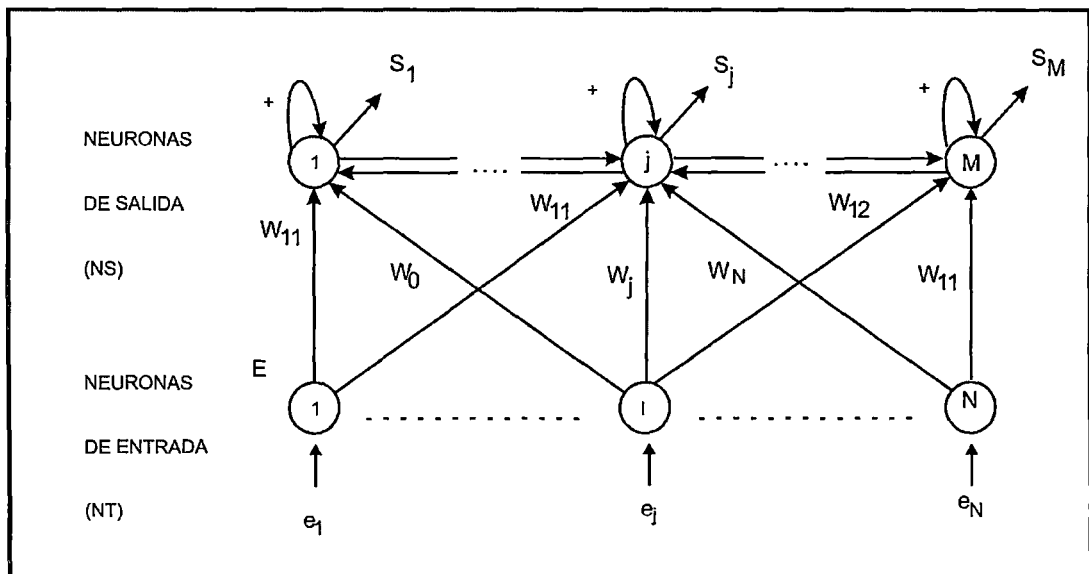


Fig. 2.7. Arquitectura de una Red de Kohonen

Cada una de las N neuronas de entrada se conecta a las M de salida a través de conexiones hacia adelante (feedforward).

Entre las neuronas de la capa de salida, puede suponerse que existen conexiones laterales de inhibición (peso negativo) implícitas, pues aunque no es-

tén conectadas, cada una de estas neuronas va a tener cierta influencia sobre sus vecinas. El valor que se asigne a los pesos de las conexiones feedforward entre las capas de entrada y salida (w_{ij}) durante el proceso de aprendizaje de la red va a depender de ésta interacción lateral.

La influencia ejercida por cada neurona sobre las demás es función de la distancia entre ellas, siendo muy pequeñas cuando están muy alejadas. Es frecuente que dicha influencia tenga la forma de un sombrero mejicano (Figura 2.8).

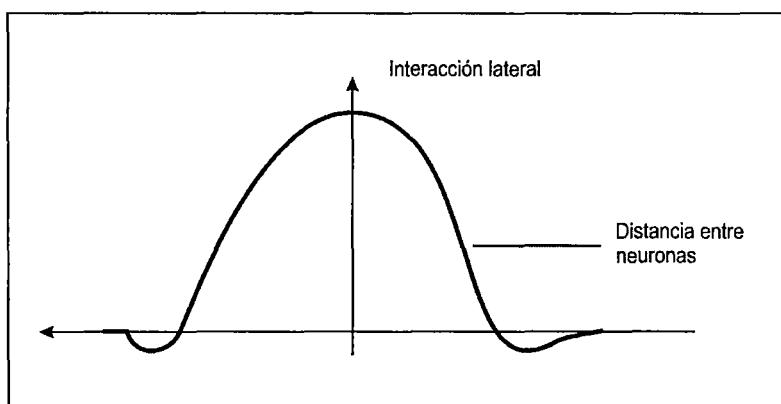


Figura 2.8. Función de interacción lateral entre neuronas.

El modelo TPM establece una correspondencia entre los datos de entrada y un espacio de salida bidimensional, creando mapas topológicos bidimensionales, de modo que ante datos de entrada con características comunes se activen neuronas situadas en zonas próximas de la capa de salida.

La interacción lateral entre neuronas de la capa de salida sigue existiendo, aunque ahora hay que entender la distancia como una zona bidimensional que existe en torno a cada neurona. Esta zona puede ser circular, hexagonal o cualquier otro polígono regular centrado en la neurona en cuestión.

2.6.3. Funcionamiento de una Red de Kohonen.

Cuando se presenta a la entrada una información $E_k = (e_1^{(k)}, e_2^{(k)}, \dots, e_N^{(k)})$, cada una de las N neuronas de la capa de salida la recibe a través de las conexiones feedforward con pesos w_{ij} . También estas neuronas reciben las entradas correspondientes a las interacciones laterales con el resto de las neuronas de salida y cuya influencia dependerá de las distancias a la que se encuentren.

La salida generada por una neurona de salida j ante una vector de entrada E_k será:

$$s_j(t+1) = f\left(\sum_{i=1}^N w_{ij} e_i^{(k)} + \sum_{p=1}^M \text{Int}_{pj} s_p(t)\right) \quad (2.26)$$

donde Int_{pj} es una función con un grafico que tiene forma de tipo sombrero mejicano que representa la influencia lateral de la neurona p sobre la neurona j . La función de activación de las neuronas de salida (f) será del tipo continuo, lineal o sigmoideal, pues esta red trabaja con valores reales

Esta red es de tipo competitivo, pues al presentarse una entrada E_k , la red evoluciona hasta alcanzar un estado estable en el que solo hay una neurona activada, la ganadora. La formulación matemática del funcionamiento de esta red puede simplificarse de la siguiente manera [Hiler José R., Martínez Victor J., 1994]:

$$s_j = \begin{cases} 1, & \text{MIN } \|E_k - W_j\| = \text{MIN} \left(\sqrt{\sum_{i=1}^N (e_i^{(k)} - w_{ij})^2} \right) \\ 0, & \text{resto} \end{cases} \quad (2.27)$$

donde $\|E_k - W_j\|$ mide la diferencia entre el vector de entrada y el vector de

pesos de las conexiones que llegan a la neurona j desde la entrada. Es en estos pesos donde se registran los datos almacenados por la red durante el aprendizaje. Durante el funcionamiento, se pretende encontrar el dato aprendido más parecido al de entrada para averiguar qué neurona se activará y en qué zona del espacio bidimensional de salida se encuentra.

Esta red realiza una tarea de clasificación ya que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información. Otra entrada parecida activa la misma neurona o una cercana a la anterior. Esto garantiza que las neuronas topológicamente cercanas sean sensibles a entradas físicamente similares. Por esta razón, la red es útil para establecer relaciones antes desconocidas entre conjuntos de datos.

2.6.4. Aprendizaje de una Red de kohonen

El aprendizaje en el modelo de kohonen es de tipo OFF LINE, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. En la etapa de aprendizaje se fijan los pesos de las conexiones feedforward entre las capas de entrada y salida.

Este tipo de red utiliza un aprendizaje no supervisado de tipo competitivo. Las neuronas de la capa de salida compiten por activarse y sólo una de ellas permanece activa ante una entrada determinada. Los pesos de las conexiones se ajustan en función de la neurona que haya resultado vencedora.

En la etapa de entrenamiento, se presenta a la red un conjunto de informaciones de entrada (vectores de entrenamiento) para que ésta establezca, en función de la semejanza entre los datos, las diferentes clases (una por neurona de salida) que servirán durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones feedforward que llegan a

cada neurona de salida se corresponderán con los valores de los componentes del vector de aprendizaje que consigue activar la neurona correspondiente. Si existiesen más vectores de entrenamiento que neuronas de salida, más de un vector deberá asociarse a la misma clase. En tal caso, los pesos se obtienen como un promedio de dichos patrones.

Durante el entrenamiento habrá que ingresar varias veces todo el juego de entrenamiento para refinar el mapa topológico de salida logrando que la red pueda realizar una clasificación más selectiva.

El algoritmo de aprendizaje es el siguiente [Hiler José R., Martínez Víctor J., 1994]:

Paso 1: Se inicializan los pesos (w_j) con valores aleatorios pequeños y se fija la zona inicial de vecindad entre las neuronas de salida.

Paso 2: A continuación se presenta a la red una información de entrada (la que debe aprender) en forma de vector $E_k = (e_1^{(k)}, \dots, e_N^{(k)})$, cuyas componentes $e_i^{(k)}$ serán valores continuos.

Paso 3: Se determina la neurona vencedora a la salida. Esta será aquella j cuyo vector de pesos W_j sea el más parecido a la información de entrada E_k . Para ello se calculan las distancias entre ambos vectores, una para cada neurona de salida. Suele utilizarse la distancia euclídea o bien la siguiente expresión similar pero sin la raíz:

$$d_j = \sum_{i=1}^N (e_i^{(k)} - w_{ij})^2, \quad 1 \leq j \leq M \quad (2.28)$$

donde:

$e_i^{(k)}$: Componente i -ésimo del vector k -ésimo de entrada.

w_{ij} : Peso de la conexión entre las neuronas i (de entrada) y j (de salida).

Paso 4: Una vez localizada la neurona vencedora j^* , se actualizan los pesos de las conexiones feedforward que llegan a dicha neurona y a sus vecinas. Con esto se consigue asociar la información de entrada con cierta zona de la capa de salida.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) [e_i^{(k)} - w_{ij}(t)] \quad \text{y en } j \in \text{Zona } j^*(t) \quad (2.29)$$

La zona $j^*(t)$ es la zona de vecindad de la neurona vencedora j^+ . El tamaño de esta zona se puede reducir en cada iteración del entrenamiento aunque en la práctica es habitual mantener esa zona fija.

El término $\alpha(t)$ es el coeficiente de aprendizaje y toma valores entre 0 y 1. Este parámetro decrece con cada iteración. De esta forma, cuando se ha presentado todo el juego de datos un gran número de veces, α tiende a cero y las variaciones de pesos son insignificantes.

$\alpha(t)$ suele tener alguna de las siguientes expresiones:

$$\alpha(t) = \frac{1}{t} \quad , \quad \alpha(t) = \alpha_1 \left(1 - \frac{t}{\alpha_2} \right) \quad (2.30)$$

2.6.5 Aplicaciones de las Redes de Kohonen

Las redes neuronales artificiales pueden ser aplicadas a situaciones en las cuales existe una relación entre variables independientes y dependientes. Desde que el fuerte de las redes neuronales es identificar patrones de comportamiento o tendencias en los datos, su uso puede ser extendidas a casos de predicción o pronóstico, validación de datos, control de procesos industriales, estudios de mercado, proyecciones de ventas, evaluación de riesgo y marketing, diagnósticos médicos, identificación de minas

submarinas, reconocimiento de voz, asignación de crédito y reconocimiento facial, controles de calidad y desciframiento del código genético.

En este trabajo de investigación, se utiliza una herramienta de libre disponibilidad, el NNclust, basada en las redes de Kohonen, para identificar patrones de consumo orientados al cliente .

CAPÍTULO III

SISTEMA CLASIFICADOR – AGRUPADOR .

MODELO DE SOLUCIÓN

3.1. Introducción

En este capítulo se presenta el sistema clasificador – agrupador cuyas componentes son: Base histórica de datos del negocio, Archivos depurados, Macros en excel (NNClust y NNClass) que proporcionan consultas y reportes finales del entrenamiento e información de salida (Patrones de consumo orientados al cliente y al producto). Ver figura 3.1.

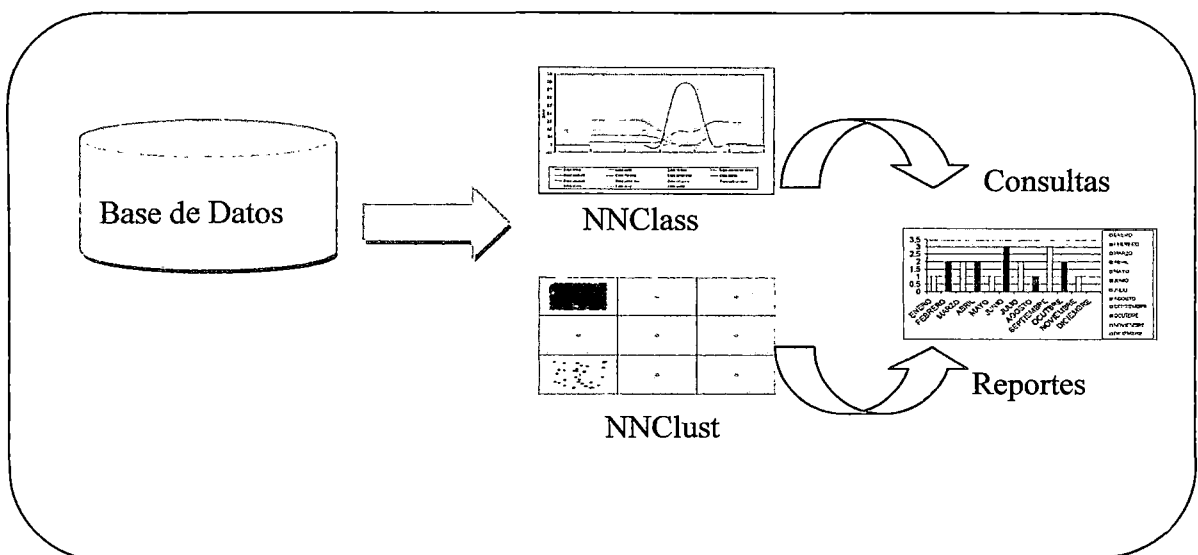


Figura 3.1 Sistema Clasificador- Agrupador

En la figura 3.2, se muestra el cuadro pictórico de los procedimientos del negocio.

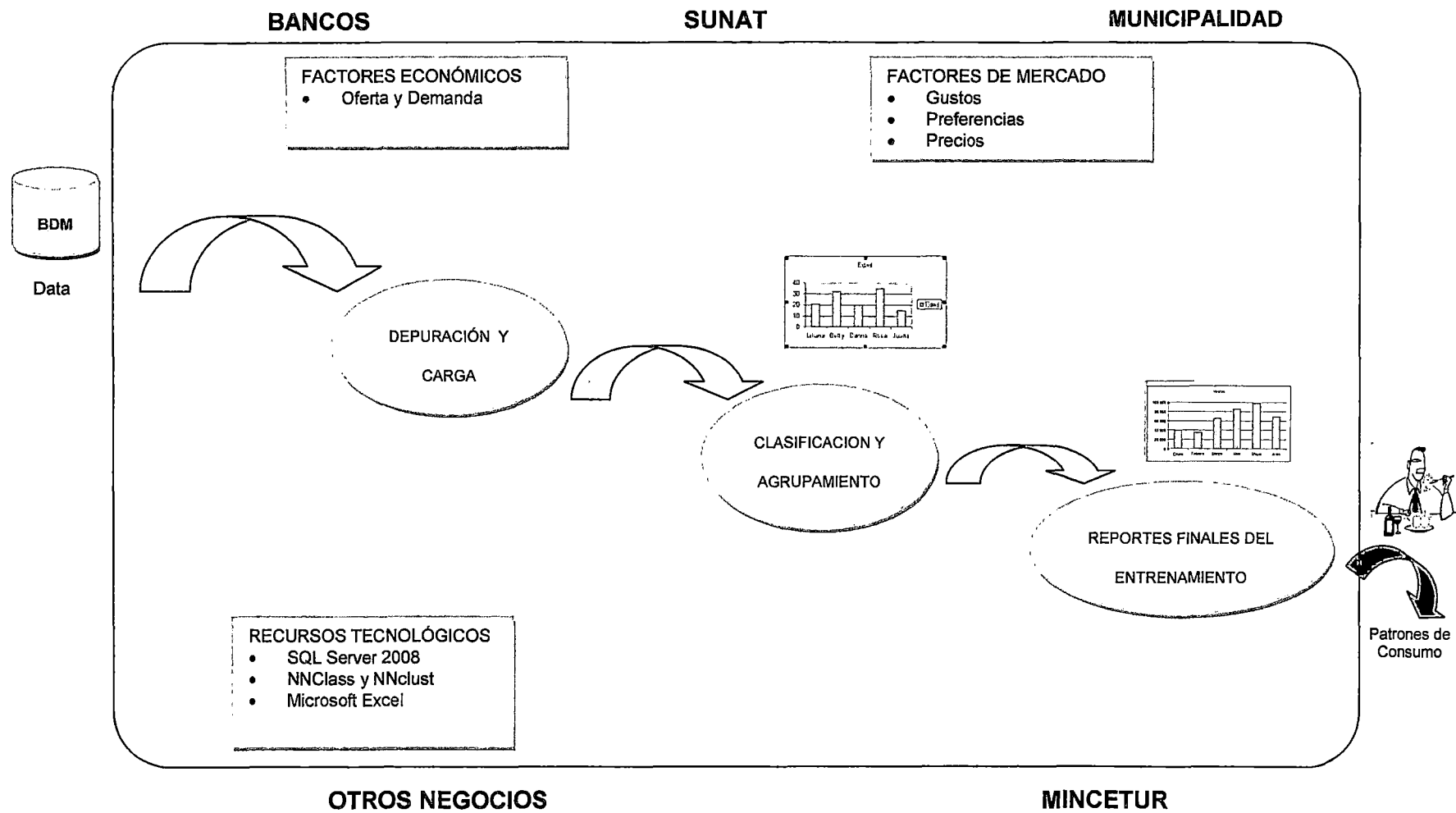


Fig 3.2. Cuadro pictórico de los procedimientos del negocio

3.2.2.1. Esquema de entrenamiento

Los datos, antes de ser entrenados, se someten a un tratamiento previo teniendo en cuenta que el valor de las variables insumo de la red pueden tener valores muy grandes y con una dispersión muy grande.

A continuación, se procede a configurar la red neuronal, esto significa indicar la cantidad de capas ocultas, número de neuronas, número de iteraciones límite para detener el entrenamiento, número de variables y número de registros de entrada, entre otros. Ver figura 3.4.

El entrenamiento propiamente dicho, empieza con la inicialización de pesos (que se ajusta internamente dependiendo de cada herramienta). Luego se efectúa el aprendizaje de la red en forma iterativa hasta tener reportes que nos permitan hacer un análisis eficiente o que el número de iteraciones

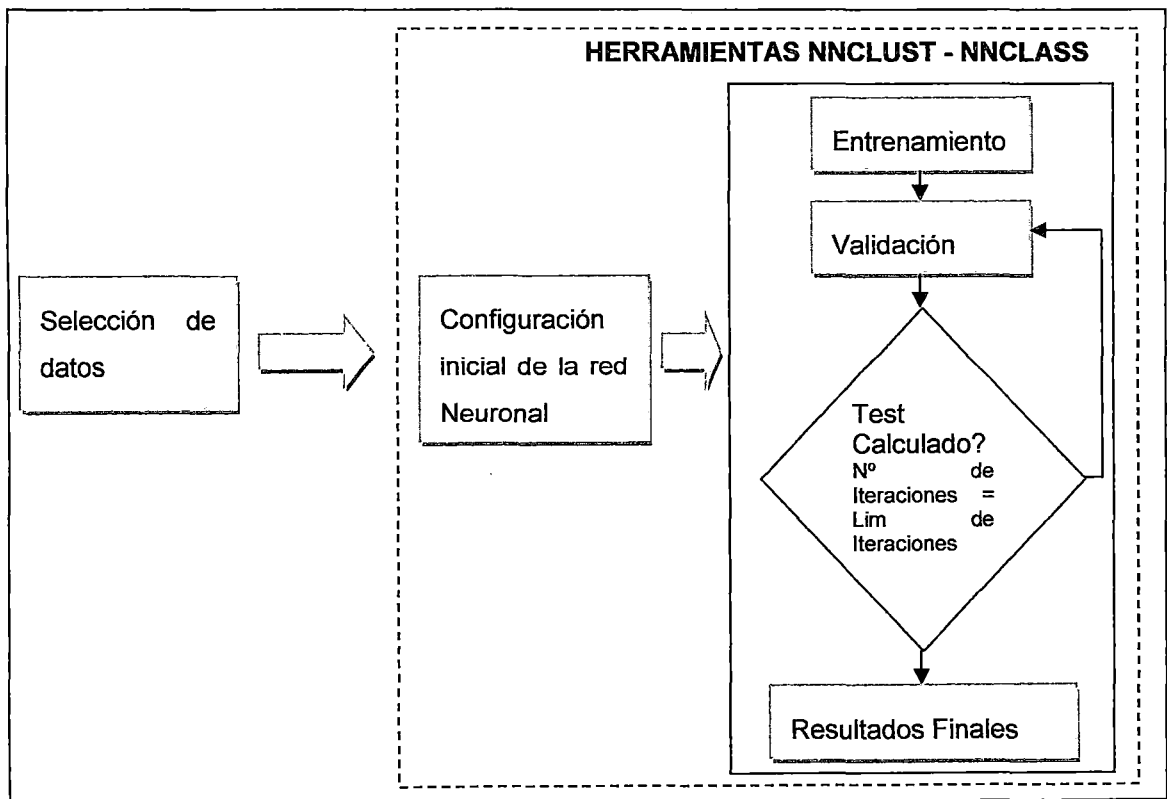


Fig. 3.4. Esquema de entrenamiento

haya llegado a la cantidad límite de iteraciones, evitando con este último criterio una iteración perpetua, en el caso no se alcance el patrón deseado. El aprendizaje es la piedra angular para el entrenamiento de la red neuronal. El aprendizaje se hace posible a través de la modificación continua del número de capas, número de neuronas, cantidad de registros y número de iteraciones.

La secuencia de aprendizaje se da en forma iterativa, procesando la información de entrada. Dentro de cada macro de entrenamiento, se realiza un recorrido hacia delante que consiste en propagar los datos a través de las neuronas de las capas intermedias (ocultas) hasta llegar a la capa de salida. Para cada neurona se calcula un valor de salida aplicando el valor de entrada a la función de activación.

Al llegar a la última capa, se calcula el patrón de esa iteración, como la diferencia entre el patrón deseado y el patrón calculado.

3.2.2.2. Esquema de Clasificación y Agrupamiento

Para realizar una clasificación, se suministra al sistema un archivo MS-Excel, con la particular característica de tener solo la hoja de entrada. En forma análoga al entrenamiento se realiza el escalamiento y la validación de los datos de entrada. Luego se procede a seleccionar de la lista de archivos de entrenamiento, uno que se haya realizado con el mismo tipo de variables cuyos valores ahora se quiere clasificar o agrupar. Ver Fig. 3.5.

El agrupamiento es un proceso sencillo, debido a que no se debe recalcular pesos, ni retropropagar valores de errores calculados. En el agrupamiento, se usan los valores del archivo de entrada, los cuales se propagan hacia delante usando los pesos finales obtenidos en el entrenamiento, de tal manera que se calculan los valores de salida a través de las capas intermedias, hasta llegar a la capa de salida. Finalizado los procesos de

clasificación y agrupamiento se crea una hoja de salida en el mismo archivo MS-Excel que se suministró con la hoja de entrada. Además, se graban los datos obtenidos, en las tablas del sistema.

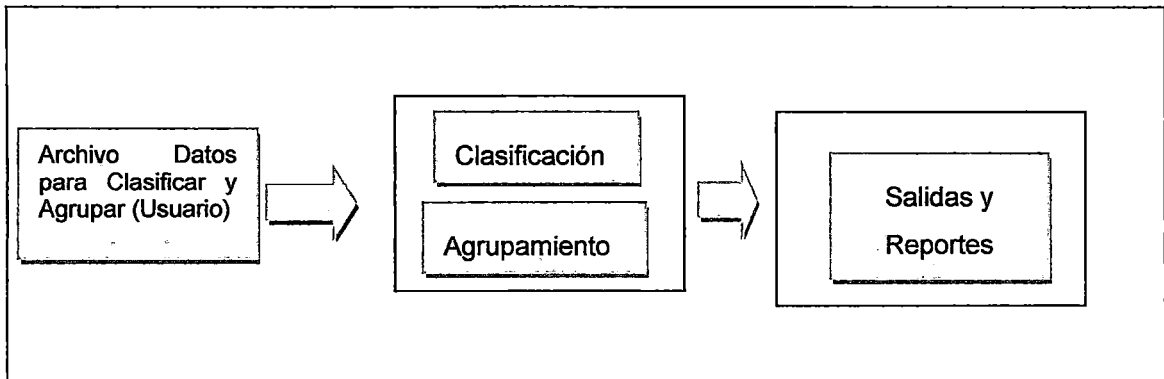


Fig. 3.5. Esquema de Clasificación y Agrupamiento

3.2.3. Salidas del sistema

Tanto el proceso de agrupación como el proceso de clasificación registran las salidas de reportes finales del sistema. Estos datos de salida pueden ser consultados a través de las opciones habilitadas con este fin.

3.2.3.1. Esquema de consultas y reportes

Concluido el entrenamiento, se puede seleccionar y visualizar los datos obtenidos en el proceso de análisis que se cargaron en la hoja de entrada del archivo MS-Excel.

Análogamente, se pueden visualizar los datos que se cargaron en la hoja de salida. Adicionalmente, para un entrenamiento específico se puede conocer la configuración de la red utilizada, es decir, el número de capas ocultas, la cantidad de neuronas para cada una de ellas, la cantidad de iteraciones, número de registros, patrones finales obtenidos y los parámetros de entrenamiento, entre otros.

3.3. Factores de innovación del sistema clasificador agrupador

a) Es multipropósito

- Puede agrupar y clasificar variables de salida de cualquier actividad que maneje información.
- Permite definir las variables con las que se va trabajar.

b) Es multidimensional

- Trabaja en forma irrestricta y dinámica con varias capas y varias neuronas.
- El usuario define interactivamente la cantidad de capas y el número de neuronas que usará el sistema.

c) Acepta múltiples formas de entrada/salida

- En línea (para configurar parámetros de la red neuronal).
- En batch (archivos de texto, archivos Excel).

d) Permite almacenamiento de datos no volátiles (permanente).

- Usa un conjunto tablas de una base de datos.
- Guarda información histórica de los procesos ejecutados.

3.4. Modelo de Solución

Para resolver el problema planteado se propone un modelo de solución que consta de cuatro fases:

- **Fase N° 1: Selección de los objetivos.-** En esta fase se estudia el problema y se decide cuál es la meta del proyecto. Asimismo se debe determinar la viabilidad del proyecto. Si se hace correctamente el planteamiento del problema, se descubren fácilmente las fuentes de datos y los algoritmos de Data Mining que se aplicarán

- **Fase N° 2: Preparación de los datos.-** Esta fase consta de tres pasos:
 - a) **Selección de los datos:** En esta parte se identifican las fuentes de datos internas o externas y se selecciona el subconjunto de datos necesarios, ya sean tablas de una base de datos o archivos de texto.
 - b) **Preproceso de los datos:** Una vez identificados los datos a utilizar, deben ser estudiados para, entender el significado de los atributos e, identificar errores de integración, pues puede ser que haya datos repetidos con distinto nombre o datos que significan lo mismo pero con diferente formato. Con el preproceso se consigue tener una data adecuada para el correcto funcionamiento de las fases posteriores del proceso.
 - c) **Transformación de los datos:** Analizado el problema y el tipo de datos disponible, se selecciona la herramienta o herramientas a aplicar. Como cada herramienta requiere un formato diferente en los datos de entrada, en esta fase se debe transformar la data de modo que se ajusten al formato de entrada de la herramienta seleccionada (NNclust y NNclass).
- **Fase N° 3: Construcción del modelo.-** En esta etapa se aplican las herramientas de análisis de datos sobre los datos transformados y preparados en las etapas anteriores. En esta etapa se utilizan las herramientas de libre disponibilidad seleccionadas (NNclust y NNclass) para buscar patrones de consumo orientados al cliente y al producto presentes en los datos. En esta fase se puede aplicar varias veces la misma herramienta o utilizar conjuntamente las dos herramientas seleccionadas.

- **Fase N° 4: Análisis de los resultados.**- En esta fase se interpretan y evalúan los resultados obtenidos en la etapa anterior. Una vez obtenidos los resultados, deben ser analizados e interpretados, y si no cumplen con las expectativas, se debe volver a aplicar las macros con otros parámetros. Esto hace que el proceso sea iterativo. En esta fase también se debe decidir cómo utilizar los resultados obtenidos.

Las tres primeras fases se desarrollan en el capítulo IV, la fase N° 4 de análisis de los resultados se desarrolla en el capítulo V.

CAPÍTULO IV

IDENTIFICACIÓN DE PATRONES DE CONSUMO EN UN SISTEMA DE NEGOCIOS UTILIZANDO REDES NEURONALES ARTIFICIALES

En este capítulo se aplica el modelo propuesto. En el capítulo anterior, se desarrollo la base de datos del negocio para identificar los patrones de consumo de Morgana – Café Lounge, utilizando Redes Neuronales Artificiales.

4.1.1. Fase N° 1: Selección de Objetivos

4.1.2. Presentación del Sistema de Negocios - Morgana Café Lounge

Morgana Café Lounge, elabora productos artesanales, con insumos de alta calidad, para obtener como resultado, productos de insuperable frescura. Cubre la necesidad de un segmento del mercado que aun desea consumir productos elaborados con insumos originales. No utiliza conservantes ni preservantes artificiales y trabaja con estrictas normas de higiene, y brinda la garantía de un producto inocuo.

En el rubro de la pastelería, los procesos requieren de tiempos largos, al reducirlos la calidad se ve afectada, debido a ello la propuesta de “Respetar los tiempos reales de producción”.

El sentido de responsabilidad hacia el consumidor, toma en cuenta la vida útil y modo de conservación de sus productos.

La diversidad de la carta, ha tomado en cuenta las estaciones del año para ofrecer un sin fin de productos selectos. Es así que con esta cantidad de beneficios e ideas concretas nace Morgana Café Lounge, para aquellos que desean deleitarse con un verdadero postre, delicado y fino, con el mejor café peruano, los mejores sándwiches clásicos y exquisitas comidas.

El personal está habilitado para trabajar en su máxima capacidad, atendiendo a 150 personas al mismo tiempo como máximo de capacidad que puede albergar el lugar. Para la atención de la clientela cuentan con una barra extensa en la que se logran colocar 10 personas, cómodamente sentadas. Dentro de la misma cuentan con 2 cantineros, y una cajera. Fuera de la Barra se cuenta con 6 saloneros o meseros, 1 seguridad y dentro de la cocina un cocinero.

El Bar-Café cuenta con un local de aprox. 400m² de distribución física en el cual el 30% representa la cocina y oficina, el resto está destinado a la terraza y el bar, esto permite recibir unas 90 personas debidamente sentadas y en la barra un capacidad de 10 personas lo cual hace un total de 100 personas. Para atender unas 300 personas por día en calidad de rotación, de Lunes a Miércoles en horarios de 4:00pm a 1:00am y de Jueves a Sábados de 4:00pm a 3:00am.

Cabe la posibilidad que en el futuro se logre expandir el negocio con miras al turismo internacional contando con otro local en área contigua dependiendo de la demanda que esta área ofrezca de aquí a cinco años.

El proyecto tiene mucha aceptación por parte del segmento de mercado al que está dirigido, debido al incremento de profesionales de cierta edad y a la escasez de este tipo de establecimientos y ligado a esto, del local con el

área bancaria considera un mercado amplio y lleno de oportunidades.

Debido a la beneficiosa posición geográfica de nuestro país, se cuenta con las tres materias primas que hacen funcionar al negocio, que son el café, licor y mariscos. Al ver la creciente demanda de café frío como bebida preferida y de moda decidieron hacerla parte del menú, de igual manera con Varela Hermanos que cuentan con una de las carteras más grandes de licores tanto nacionales como internacionales y contando con ingenios azucareros; y por último punto los mariscos que representan a la costa y beneficia grandemente. Al negocio se puede agregar la variedad de acompañamientos que se puede unir a los platos producto del sector agrícola, todo esto trae como consecuencia el abaratamiento de los costos.

Observando lo que indican las encuestas realizadas por la empresa, se considera que el local cuenta con el área perfecta y el tamaño adecuado para pensar en proyecciones de crecimiento en el mediano plazo debido a la creciente demanda.

La empresa cuenta con maquinaria de bar y cocina no de primera mano pues se considera, que es posible obtener maquinaria de segunda mano con mucha vida útil por delante a la mitad de su costo original y conforme avance se actualice el stock de maquinarias. Adicionalmente a esto, muchos proveedores ofrecen su maquinaria con tal de comprometer al consumo mensual de sus productos y la promoción de los productos dentro del local.

Dentro del proceso el equipo a utilizar es de uso común en restaurantes y bares, por consiguiente no se considera que exista ningún tipo de inconveniente en la disponibilidad local de mismo. El flujo de proceso (ver figura 4.1) de la empresa es del tipo intermitente secuenciado debido a que el cliente escoge un determinado aperitivo o bebida a consumir, o sea, se desarrolla un orden exacto o secuencia de procesamiento a medida que las

ordenes llegan y pasan las estaciones

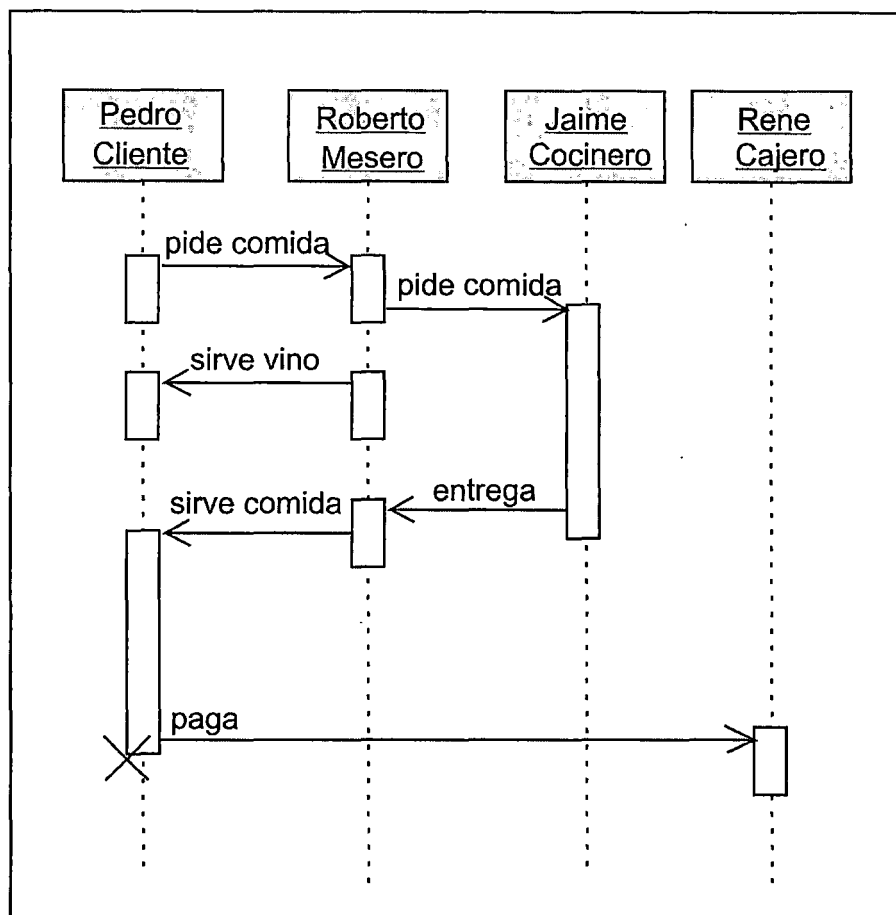


Fig. 4.1. Presentación del Proceso del Sistema de Negocio - Morgana Café Lounge

El local de Morgana Café está localizado en la Calle General Suárez 205 Miraflores – Lima 18 (ver figura 4.2.).

La ubicación del local cuenta con las siguientes características:

- Está situado a media cuadra de Tomas Marsano.
- Se encuentra enclavado en una zona residencial de alto valor, cuyos residentes pertenecen a un nivel socio económico alto.

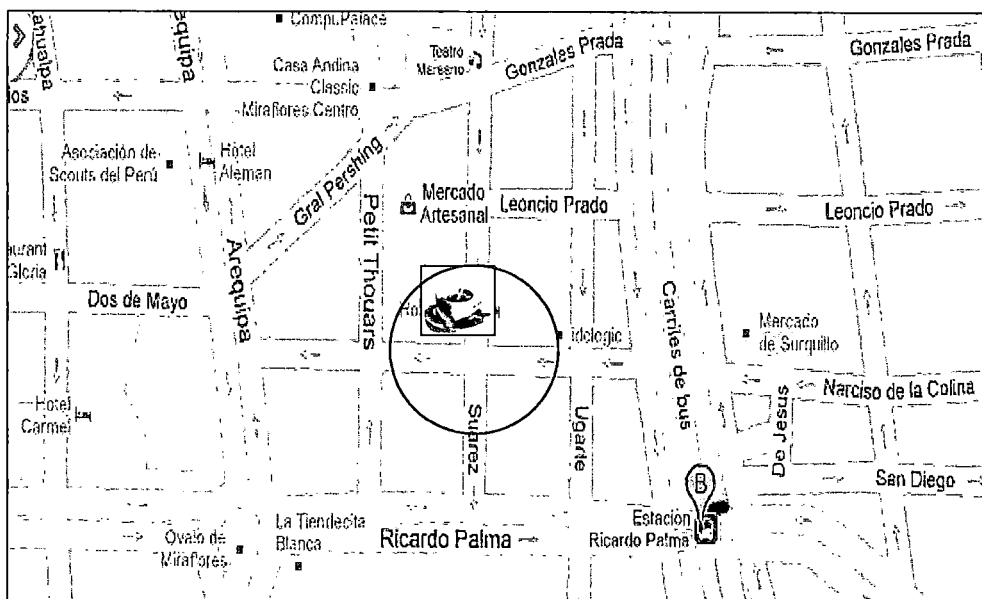


Fig. 4.2. Ubicación del local

- Está próximo a las Avenidas Dos de Mayo, Arequipa y General Pershing, que constituyen importantes áreas comerciales de la zona.
- Se encuentra cerca a varios mercados artesanales de la zona.
- Se encuentra a una cuadra del Ovalo de Miraflores, donde hay una iglesia, varios museos, y una gran concentración de comercios.
- A menos de 2 cuadras varios hoteles, el teatro Marsano, varias salas de cine.

Como se puede notar, el local tiene una ubicación privilegiada (como se observa en la figura 4.2). El hecho que hayan mercados artesanales aseguran un gran flujo de personas por la zona.

4.1.3. Identificación de los Objetivos

El uso de la informática en la Empresa Morgana Café Lounge ha simplificado y acelerado muchos procesos internos, generando gran cantidad de información en la base de datos. Sin embargo, estos datos contienen valiosa información sobre el negocio, que podría ser utilizada para generar mayor rentabilidad y ganancias.

El problema que se intenta solucionar es el del aprovechamiento deficiente de la información generada en la empresa. Como objetivo principal, se buscará estudiar la demanda de Morgana – Café para optimizar la demanda en función de las necesidades del cliente.

En este trabajo se trata de mostrar que no es necesario poseer grandes capitales, ni equipos sofisticados, ni contar con expertos en estadística, ni tener una gran empresa para poder aprovechar la información, pues para todo negocio que posea una computadora de escritorio con acceso a internet, y mínimos conocimientos de base de datos, puede aplicar el modelo propuesto.

Para aplicar el modelo propuesto, se utilizó una base de datos de consumo proporcionada. Se ha aplicado redes neuronales artificiales, combinando herramientas de libre disponibilidad para detectar patrones de consumo y segmentar a la clientela. A través de este análisis y aplicando el modelo propuesto en el capítulo anterior se busca proponer modificaciones y mejoras a la oferta, para adecuarla a la demanda y las necesidades implícitas de los clientes.

4.2. Fase N° 2: Preparación de los Datos

En la preparación de los datos se consideraron los siguientes pasos:

- Selección de los datos
- Pre proceso de los datos
- Transformación de los datos

4.2.1. Selección de los datos

En esta parte, se construyó el modelo conceptual, el modelo lógico y el modelo físico de la base de datos y se realizó la selección de los datos de Morgana Café (Ver figura 4.3).

4.2.2. Pre Proceso de los Datos

Se identificaron los atributos de las tablas seleccionadas para tener un conjunto adecuado de datos para el correcto funcionamiento de las fases del proceso.

En el pre proceso de los datos se obtuvo la siguiente tabla de atributos (ver figura 4.1) con su descripción correspondiente.

4.2.3. Transformación de los datos

En esta fase se transformaron los datos de modo que se ajustaron al formato de entrada de algoritmos seleccionados.

Selección de Entidades de la Base de Datos MORGANA CAFÉ para el Modelado en NNClust y NNClass

Las Entidades que se involucraron para el modelado de datos son: Producto, Cliente, Empleado, Pedido y, Comprobante.

Entidad	Detalle
Id_Producto	Posee un identificador del tipo de producto.
Cantidad	Cantidad de producto vendido.
Fecha	Fecha en que se realiza la instancia venta.
Hora	Hora en que se realiza la instancia venta.
Id_DetalleFactura	Identificador de cada boleto.
Precio	Costo unitario del producto.
Total de Factura	Registra el total a pagar.
Código Categoría	Especifica la categoría de los productos.
Id_Empleado	Identificador del personal de servicios.
Descripción	Describe detalles secundarios.

Tabla 4.1. Pre Proceso de los Datos

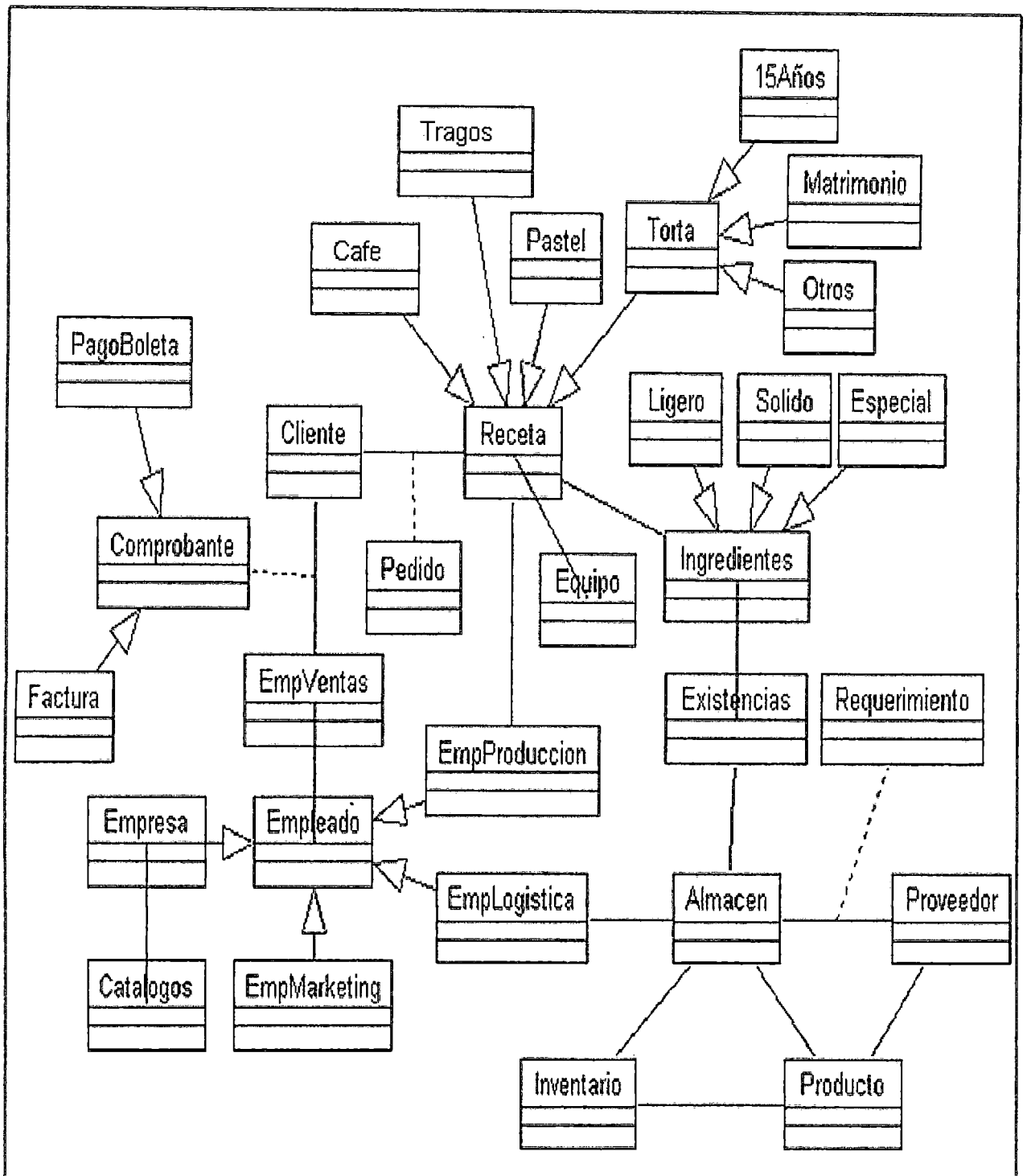


Fig. 4.3. Modelo Conceptual de la Base de Datos de Morgana Café - Lounge

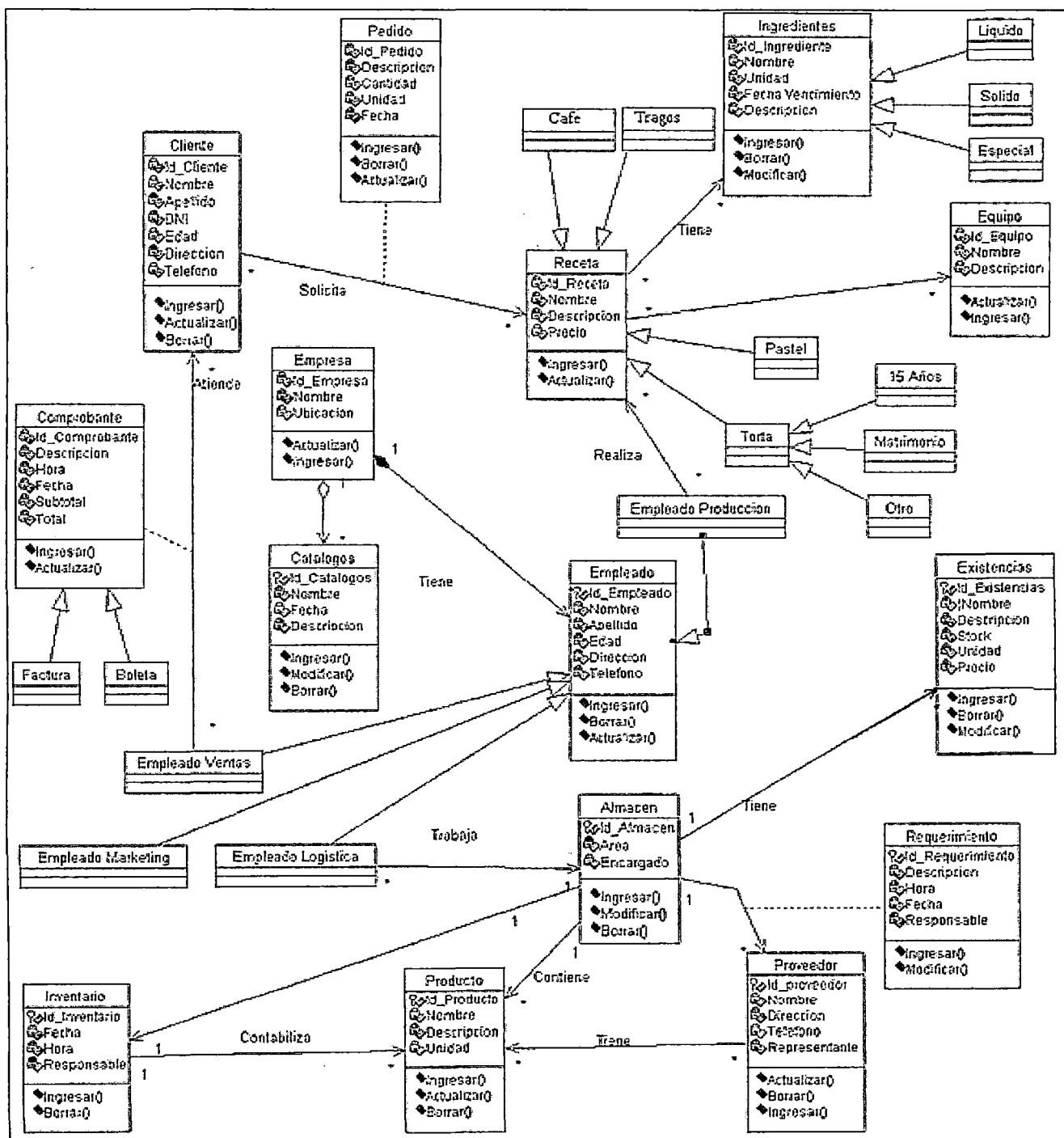


Fig. 4.4. Modelo Lógico de la Base de Datos de Morgana Café - Lounge

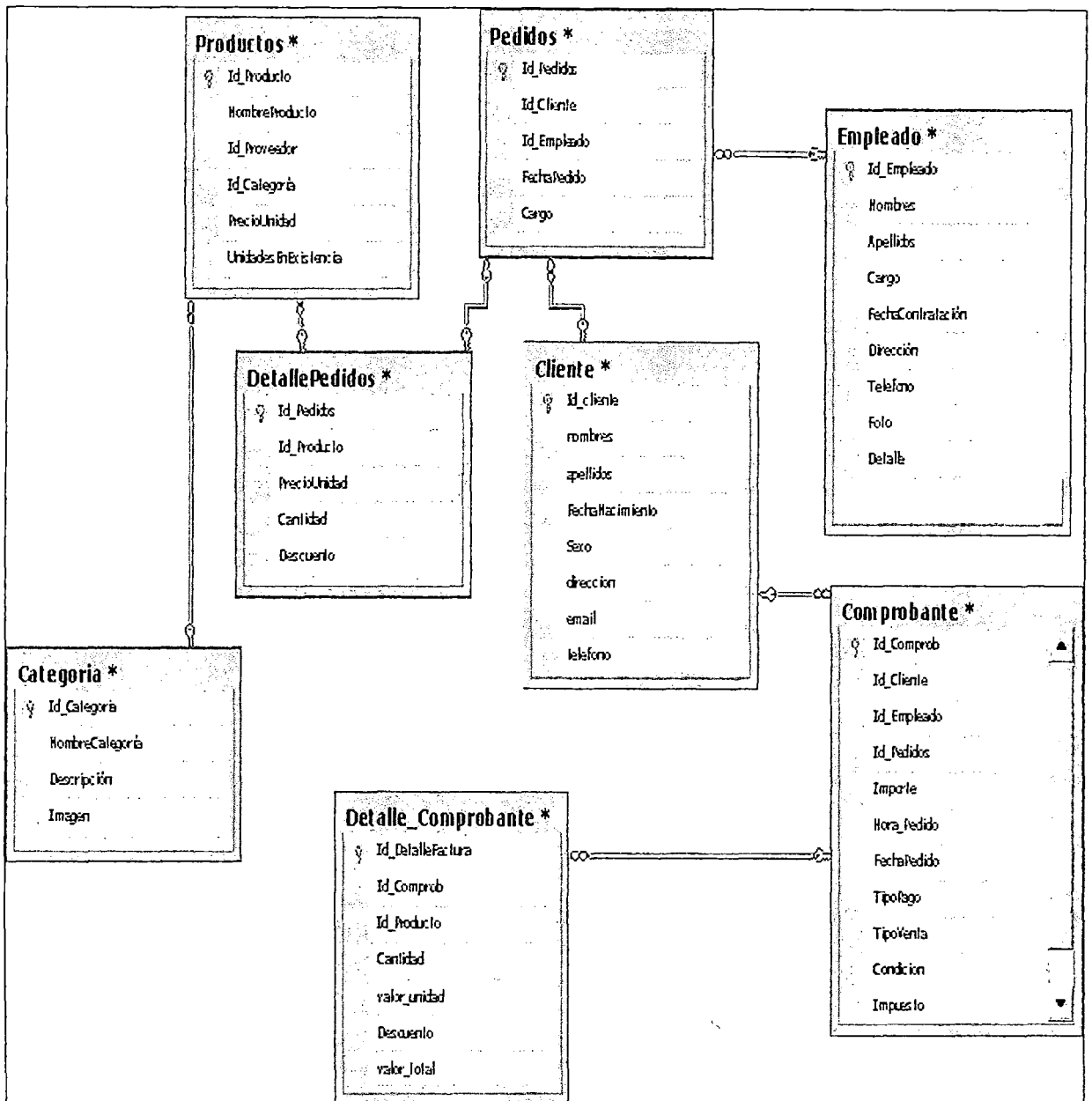


Fig. 4.5. Modelo Físico de la Base de datos de Morgana Café - Lounge

De ellos los campos seleccionados son:

De Producto:

- Código de Producto.
- Nombre de Producto
- Cantidad.
- Precio.

De Cliente:

- Identificador del Cliente.
- Sexo del Cliente.
- Edad de Cliente

Id_cliente	nombres	apellidos	FechaNaci	Sexo	direccion	email	telefono
ak	Maria	Salas Contre	10/12/1984	F	Av Dos de M	mary@hotmail.co	2458796
an	Ana	Trujillo Verd	06/08/1986	F	Av Benavide	Nany@gmail.com	2784987
Anton	Antonio	Moreno Tapi	02/12/1987	M	Av Dos de M	Ant@hotmail.com	3568794
Car	Carmen	Mendoza Ma	03/11/1991	F	Av Petit Tho	Cari@yahoo.es	2487659
Jose	Josue	Merino Loza	14/10/1980	M	Av Leoncio P	Jos@hotmail.com	997865242
jos	José	Marquez Mo	16/05/1984	M	Av Benavide	Jos@gmail.com	3458765
Kor	Korina	Castro Zevall	20/12/1978	F	Av Ricardo P	Kor@hotmail.com	2457897
Lu	Luis	Flores Mene	29/07/1983	M	Av Arequipa	Lu@hotmail.com	2548796
Mar	Malena	Rios Solis	17/02/1982	F	Av Narciso d	mar2gmail.com	7854876
Meli	Melisa	Diaz Contrer	03/11/1987	F	Gonzales Pra	Meli@yahoo.es	2785498
Ceci	Cecilia	Ordoñez Cav	10/06/1985	F	Av Benavide	cecy@hotmail.com	988784652

Tabla 4.2. Transformación de datos

De Empleado:

- Identificador del Empleado.
- Cargo.

Id Emple	Nombres	Apellidos	FechaNaci	Sexo	Cargo	FechaContratac	Dirección	telefono	foto	Notas
1	Carmen	Salazar Rojas	12/10/1985	F	Mesera	07/09/2010 00:00	Av Armenda	2457896		
2	Miguel	Contreras M	19/06/1986	M	Cocinero	10/12/2009 00:00	Jr Bolivar N°	345687		
3	Sandra	Meneses Cal	22/01/1980	f	Mesera	20/06/2010 00:00	Jr Tupac Am	345687		

Tabla 4.3 Transformación de datos

De Pedido:

- Fecha de pedido.

De Factura:

- Identificador de Factura.
- Total a pagar.

Id. Detalle Fact	Id. Factura	Id. Producto	Cantidad	valor unidad	valor tot
1	1	1	3	8	24
2	2	1	2	8	16
3	3	2	4	7	28

Tabla 4.4 Transformación de datos

4.3. Fase N° 3: Construcción del Modelo

En esta fase se definió el modelo de RNA para el sistema de la empresa Morgana Café Lounge que permitió identificar patrones de consumo en el sistema de negocio utilizando redes neuronales.

En primer lugar, se definió un modelo para un sistema de negocio competitivo utilizando indicadores económicos. Para esto se utilizó como referencia la Tesis de Gomes de Lima [Manuel Sebastián Gómez de Lima, Redes Neuronales Aplicadas a los Negocios 2007], donde se aplica técnicas de redes neuronales para identificar patrones en una dulcería.

Definición del Modelo de RNA

Para resolver el problema de identificación de patrones de la empresa buscando minimizar la cota de error, se ha considerado:

- El tipo de técnicas que se va a aplicar
- El tiempo de respuesta requerida.
- Los equipos informáticos necesarios para la información.

Esto nos permitió elegir el modelo en cual se considera los mapas autoorganizados de Kohonen y la red Backpropagation que cumplen los requisitos para su aplicación (ver capítulo 3)

La selección del modelo se realizó una vez especificado a detalle las características del problema de investigación, donde se eligió un modelo apropiado de RNA con el cual se comenzó a hacer las pruebas.

Para poder solucionar el problema de la aplicación de las redes neuronales a la base de datos de la empresa Morgana Café Lounge, fue necesario:

- Adecuar la base de datos para posibilitar la correcta aplicación de las herramientas. Una vez definidas las herramientas y depurada la base de datos, se aplicarán las herramientas.
- Reuniones con dueños y empleados del local. De esta forma se podrá evaluar el desempeño de las herramientas en función de los resultados, y se podrán estudiar propuestas de mejora para la oferta local.

Las herramientas para el análisis de los datos fueron programas NNClass y NNClust. Estas aplicaciones son plantillas en Excel que ejecutan un conjunto de macros. Esto facilita la compatibilidad con otros programas, lo cual simplifica la implementación y el análisis de la información. La elección de estas dos herramientas reside, en que son software de distribución gratuita a través de Internet. Existen otros programas de minería de datos y Redes Neuronales, pero la mayoría son costosos o son muy específicos.

Las dos herramientas mencionadas se pueden descargar de Internet a través de la página web de su creador Angshuman Saha 2006:

<http://www.geocitrus.com/adotsaha/index.html>

Allí también se encuentran descripciones de las herramientas, así como un listado de propiedades y limitaciones que presentan más adelante.

La fase N° 4: Análisis de los Resultados se presenta en el capítulo V.

CAPÍTULO V

ANÁLISIS DE LOS RESULTADOS

En este capítulo se analizan los resultados obtenidos en la fase anterior. El análisis de los datos lo haremos en dos etapas: Estudio de la segmentación y estudio de la clasificación.

5.1. Etapa N° 2: Estudio de Clasificación

5.1.1. Aplicación de la Herramienta: NNClass

El NN Class es una herramienta basada en el modelo Backpropagation para encontrar patrones de consumo en las franjas horarias que se han considerado. Con esta finalidad se tomaron los datos de la tabla 5.1 que contiene 72 productos clasificados en 38 rubros. La información ya se encontraba depurada. La cabecera de dicha tabla es como sigue:

IDENTIFICADOR	PRODUCTO	FRANJA	CANTIDAD	CLIENTE	MOZO
6	Jarra	2	2	1	2
6	Jarra	2	2	1	1
7	Café Morgana	2	2	1	1
7	Café Morgana	2	2	1	2

Donde las variables: CANTIDAD oscila entre 1 y 12, FRANJA, CLIENTE y MOZO las variables han sido codificadas de la siguiente manera:

FRANJA	DENOMINACIÓN
1	Mañana (8 a 10 a.m.)
2	Media mañana (10 a 12.m.)
3	Medio día (12 a 3 p.m.)
4	Tarde (3 a 6 p.m.)
5	Noche (6 a 10 p.m.)
6	Tras noche (10 a 12 p.m.)

ódigo	Rubro	Código	Producto	Precio
2	Café Morgana	1	Café Morgana	9.00
3	Café Espresso	2	Café Espresso	4.00
4	Doble	3	Café tamaño doble	10.00
5	Café con Leche	4	Café con Leche	8.00
6	Te	5	Te	3.00
8	Capuccino Italiano	6	Café Capuccino Italiano	7.00
9	Capuccino	7	Café Capuccino	7.00
10	Adicionales	8	Gelatinas	4.00
10	Adicionales	9	Panes	2.00
11	Tragos con Café	10	Café Cubano	6.00
11	Tragos con Café	11	Café Irlandés con vainilla	8.00
14	Bebidas s/ Alcohol	12	Agua Mineral	4.00
14	Bebidas s/ Alcohol	13	Kero	4.00
14	Bebidas s/ Alcohol	14	Chicha morada	2.50
14	Bebidas s/ Alcohol	15	Gaseosas	4.00
16	Dulces	16	Mazamorra morada	5.00
16	Dulces	17	Arroz sambito	6.00
16	Dulces	18	Arroz con leche	5.00
18	Tortas	19	Suspiro a la limeña	6.50
18	Tortas	20	Maracumango	6.50
18	Tortas	21	Trufa de lúcuma	8.00
18	Tortas	22	Crema volteada	6.00
18	Tortas	23	Selva negra	8.00
18	Tortas	24	Cheesecake	8.00
18	Tortas	25	Tres Leches	6.00
18	Tortas	26	Pie de Limón	5.50
18	Tortas	27	Torta de chocolate	5.00
21	Sandwiches	28	Tres quesos	10.00
21	Sandwiches	29	Butifarra	10.00
21	Sandwiches	30	Relleno de calamares	15.00
21	Sandwiches	31	Especial de casa	10.00
21	Sandwiches	32	Mixto	7.00
21	Sandwiches	33	Croissant de Pollo	7.00
21	Sandwiches	34	Croissant Mixto	7.00
21	Sandwiches	35	Combo de sándwich	18.00
22	Plato especial del día	36	Plato especial del día	15.00
23	Brochetas	37	Brochetas	3.50
24	Empanadas	38	Empanada de carne	5.50
24	Empanadas	39	Empanada de pollo	5.50
25	Tartas	40	Tarta de Espinacas	5.00

26	Tortillas	41	Tortillas	6.00
27	Ensaladas	42	Ensalada de espinaca	5.00
27	Ensaladas	43	Ensalada Mixta	4.00
27	Ensaladas	44	Ensalada de Lechugas	3.00
32	Pizza Hawaina	45	Pizza Hawaina	4.50
33	Café pasado	46	Café pasado	5.50
36	Alfajores	47	Alfajor	2.50
37	Bebidas c/ Alcohol	48	Vinos Tabernero	33.00
37	Bebidas c/ Alcohol	49	Cerveza Cusqueña	8.00
38	Yogurt	50	Yogurt	5.50
40	Carnes	51	Lomo Saltado	17.00
40	Carnes	52	Bistec a lo pobre	18.00
41	Ensalada Rusa	53	Ensalada Rusa	8.50
42	Jugos	54	Jugo surtido	8.50
42	Jugos	55	Jugo Lúcuma	7.00
42	Jugos	56	Jugo surtido	6.00
43	Pastas	57	Espagueti en salsa 3 quesos	16.00
43	Pastas	58	Pasta a la italiana	18.00
44	Pescados	59	Chita Frita	25.00
44	Pescados	60	Pescado Frito	15.00
45	Pollo	61	Pollo a la plancha	13.00
45	Pollo	62	Pollo a la parrilla	14.00
45	Pollo	63	Milanesa de Pollo	12.00
46	Antipasto	64	Antipasto	8.00
47	Rosquillas	65	Rosquillas	3.00
48	Ensalada de Frutas	66	Ensalada de Frutas	5.00
49	Desayunos	67	Desayunos	10.00
50	Sopas	68	Sopa especial	5.00
50	Sopas	69	Caldo de Gallina	6.00
51	Milkshakes	70	Milkshakes	8.00
52	Helados	71	Helados	5.00
54	Menú Light	72	Menú Light	8.00

Tabla 5.1: Listado de Rubros y Productos

Pero se usó esencialmente la segunda columna, con las variables tal como aparece en la cabecera de la data.

FRANJA	DENOMINACIÓN
1	Mañana (8 a 10 a.m.)
2	Media_mañana (10 a 12.m.)
3	Medio_día (12 a 3 p.m.)
4	Tarde (3 a 6 p.m.)
5	Noche (6 a 10 p.m.)
6	Tras_noche (10 a 12 p.m.)

En la pestaña User Input de la herramienta NNClass se completaron las celdas concernientes al número de variables, número de capas ocultas, los parámetros de entrenamiento tal como indica la siguiente imagen.

Luego haciendo click en la pestaña Data se ingresan los datos, tal como muestra la figura 5.1:

85	Enter your Data in this sheet						
86	Instructions: Start Entering your data from cell AC105.		Specify variable name in row 103.				
87	Make sure that the row 104 is blank.						
88	Specify variable type in row 102.						
89	Cont - for continuous Input,						
90	Cat - for Categorical Input,						
91	Output -for Output var.						
92	Omit - if you don't want to use the variable in the n						
93	Specify variable names in row 103						
94	For each continuous Input, there will be 1 neuron in Input Layer.						
95	For Each categorical Input with K levels, there will be K neurons in Input Layer						
96	Please make sure that there are <u>no more than 50 neurons</u> in Input Layer.						
97	There should be <u>exactly 1 Output variable</u> - application will treat it as Categorical.						
98	There should be <u>no more than 40</u> Categorical Variables.						
99							
100							
101							
102	Var Type	Omit	Output	Cont	Cont	Cont	Cont
103	Var Name	IDENTIFICA	PRODUCTO	FRANJA	CANTIDAD	CLIENTE	MOZO
105			6 pocillo	2	2	1	2 A
106			6 pocillo	2	2	1	1 B
107			7 Café Morgana	2	2	1	1 C
108			7 Café Morgana	2	2	1	2 A
109			7 Café Morgana	2	2	1	2 B
110			6 pocillo	2	2	1	1 C
111			7 Café Morgana	2	2	1	1 A
112			6 pocillo	2	2	1	2 B

Fig. 5.1 Interfaz Gráfica de NNClass (Data)

Luego de correr la herramienta con los datos y todos los parámetros definidos, se obtuvo los siguientes resultados:

Una tabla con las matrices de pesos obtenidos por la red luego de haber terminado su fase de entrenamiento, validación y test.

Neural Network Model for Classification		Created On: 27-dic-10				
% MissClass.(Training)	69.00%	% MissClass.(Validation)				
Number of Hidden Layers	2					
Layer Sizes	4	15	5	15		
True Output (if available)	merienda					
Model Output						
Raw Input		Cont	Cont	Cont	Cont	
	Bias	FRANJA	CANTIDAD	CLIENTE	MOZO	
	1	3.7210	10.9000	1.7380	1.4670	
Transformed Input	Bias	FRANJA	D	CLIENTE	MOZO	
	1	0.5442	0.8999	0.2461	0.4670	
Hdn1_bias	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Hdn1_Nrn1	4.9584	-28.1124	6.6433	-0.8665	-15.9154	-12.0074
Hdn1_Nrn2	5.1166	-24.5650	20.4610	-3.5503	-1.1386	8.7565
Hdn1_Nrn3	-1.6201	-16.1709	7.2333	-0.4135	-1.2388	-4.5910
Hdn1_Nrn4	9.7442	5.7026	-16.4125	10.0806	0.9199	0.9875
Hdn1_Nrn5	-20.7877	9.1606	10.3818	-8.4912	-3.5493	-10.2066
Hdn1_Nrn6	-4.2842	33.2926	-15.0000	10.6938	-2.8598	1.6302
Hdn1_Nrn7	-20.2013	14.8960	-1.3733	-2.8036	-3.0455	-15.4430
Hdn1_Nrn8	-13.7757	7.0072	1.2188	-3.7571	-2.2947	-10.8617
Hdn1_Nrn9	1.0478	-36.2535	27.4181	-9.7778	-8.2650	-0.2726
Hdn1_Nrn10	-5.2893	10.3367	33.2277	-6.7482	0.0602	28.6059
Hdn1_Nrn11	-17.1608	26.4948	9.7013	-4.9855	-0.1539	4.6892
Hdn1_Nrn12	12.5287	1.3484	-17.9647	6.8818	1.6591	-0.4363
Hdn1_Nrn13	-11.4929	5.8268	-2.6595	1.1313	-2.4507	-11.5814
Hdn1_Nrn14	-5.9996	-3.1689	1.0085	-0.4758	2.5234	-5.7551
Hdn1_Nrn15	5.1470	-1.6644	-0.3367	0.1438	1.9501	4.8844

Enter your inputs in the range AG112:AJ112 - the cells marked in green.

Fig. 5.2 Interfaz Gráfica de NNClass (Calc)

	1.0000	0.0000	0.9998	0.0100	0.7286	0.0000	0.8362	0.0000	0.0000	0.4323	1.0000	0.9909	0.3926	0.0000	0.0032	0.9925
Hdn2_bias	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Hdn2_Nrn1	-1.8056	-4.0054	1.7515	-1.8142	-3.9348	1.3591	3.6535	1.7504	0.4155	-4.2658	4.9108	1.9106	-4.1828	1.9827	-0.1036	0.0930
Hdn2_Nrn2	0.1135	0.3096	-0.6806	-0.7324	-0.0446	0.9600	1.6645	3.8648	-1.1655	-3.7548	-4.1352	-2.8107	-0.3160	2.7652	0.1490	0.4834
Hdn2_Nrn3	-0.0991	2.7648	2.0950	-1.5557	0.1360	-2.3956	-2.5935	-7.1883	-4.0962	4.3896	-6.5677	-1.8602	3.9572	-0.8427	0.1454	0.8438
Hdn2_Nrn4	-6.2834	-5.1595	-3.7634	-3.1853	1.2683	3.2019	5.1039	2.1321	-0.1994	-3.0880	-4.5450	3.1005	-0.6217	0.9175	0.1238	-1.1665
Hdn2_Nrn5	-0.5196	5.0227	2.9190	2.6521	1.4584	-3.2290	-3.2047	-5.5406	-2.3852	3.3169	1.2817	-4.7693	1.1494	-0.8766	0.6188	0.4342

Tabla 5.2.a Listado de Resultados

	1.0000	0.9617	0.0006	0.0153	0.0001	0.4206
Op_bias	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Op_Nrn1	-3.1836	-2.4124	-1.3353	0.4410	-2.1275	-0.0139
Op_Nrn2	-3.4309	-0.5667	-0.2272	-0.2623	-1.0452	0.3446
Op_Nrn3	-4.7457	-5.8621	-2.1353	8.1866	-6.0603	-0.4469
Op_Nrn4	-3.9177	-0.9324	-1.4847	-0.5786	-3.3536	1.6905
Op_Nrn5	-4.9542	-3.1156	-1.2576	-9.3922	-3.1102	5.9938
Op_Nrn6	-3.9878	0.1076	-0.5248	-1.8275	6.6730	-1.8964
Op_Nrn7	-3.1482	-1.5755	-0.2131	-0.0889	-1.2375	-0.6000
Op_Nrn8	-3.6329	-0.4753	-0.5665	0.8664	-1.0610	0.0781
Op_Nrn9	-2.9616	-3.7831	-0.2248	2.4066	0.4828	-3.3706
Op_Nrn10	-3.1435	2.5430	-0.9593	-1.3740	-4.8875	1.1252
Op_Nrn11	-2.0870	-3.9827	-4.1876	-4.9875	-9.0231	3.2294
Op_Nrn12	-4.8578	-2.9784	2.9399	-1.5791	-1.4161	1.7097
Op_Nrn13	0.5598	3.5625	1.5573	-8.1671	-2.5622	-8.6760
Op_Nrn14	-6.4670	-4.4412	2.9355	0.1840	2.7224	1.1524
Op_Nrn15	-4.6723	0.6220	-1.3444	-2.8294	1.5715	1.5913

Tabla 5.2.b Listado de resultados

La curva de entrenamiento, indica que se ha conseguido con un 69% del total de la 100 iteraciones que se habían impuesto a la red.

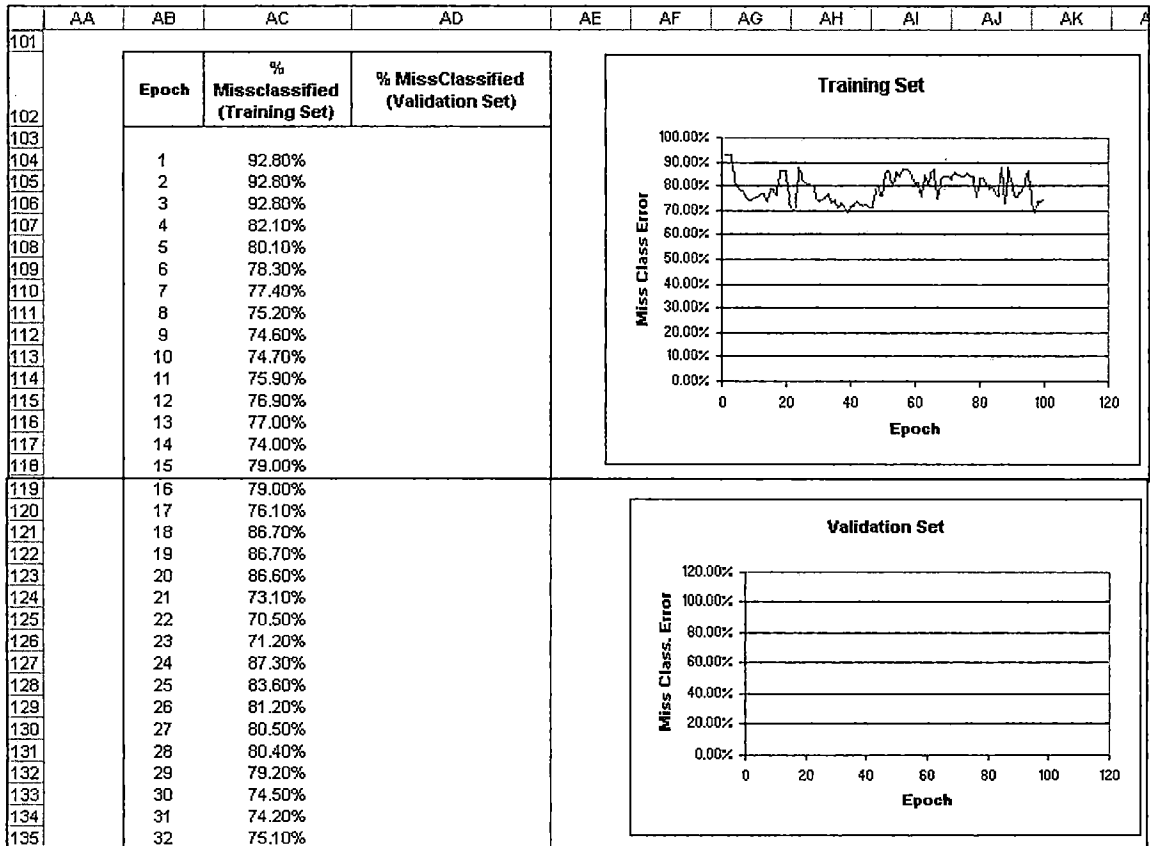


Fig. 5.3 Interfaz Gráfica de NNClass (Output)

El resultado es que se consiguieron 15 clases categóricas, que son los productos más representativos en la data.

Category Table	
PRODUCTO	
15	
1	Jarra
2	café morgana
3	desayuno
4	café con leche
5	pescados
6	pastas
7	capuccino
8	dulces
9	sandwiches
10	helados
11	sopas
12	carnes
13	merienda
14	café al paso
15	bebidas con alcohol

5.1.2. Resultados obtenidos al final del análisis

Con la pestaña Profile, se obtienen los perfiles de las 4 variables antes indicadas. Estos perfiles son las curvas que la herramienta obtiene al variando solo uno de las variables y mostrando su recorrido sobre una superficie 4-dimensional. Esta función multidimensional es la que en la fase del entrenamiento es recompilación, hasta encontrar un mínimo local.

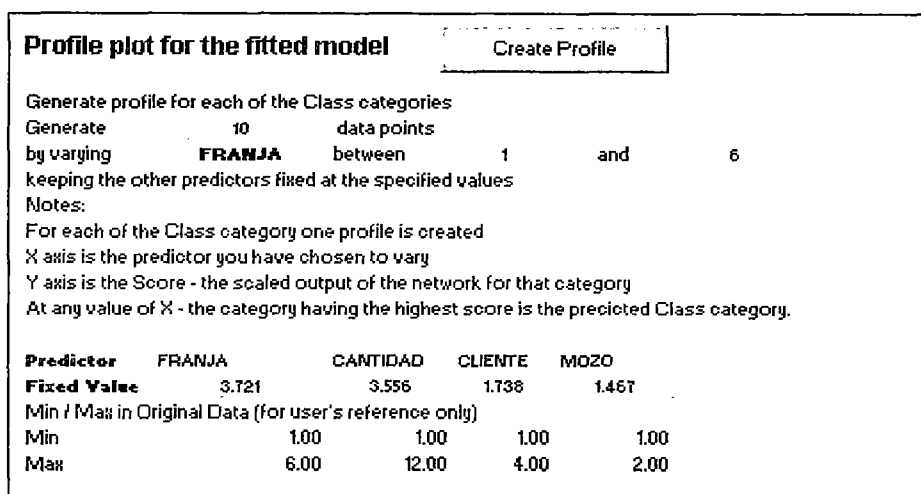


Fig. 4. Interfaz Gráfica de NNClass variable franja

Variable FRANJA: El perfil que presenta la que sigue:

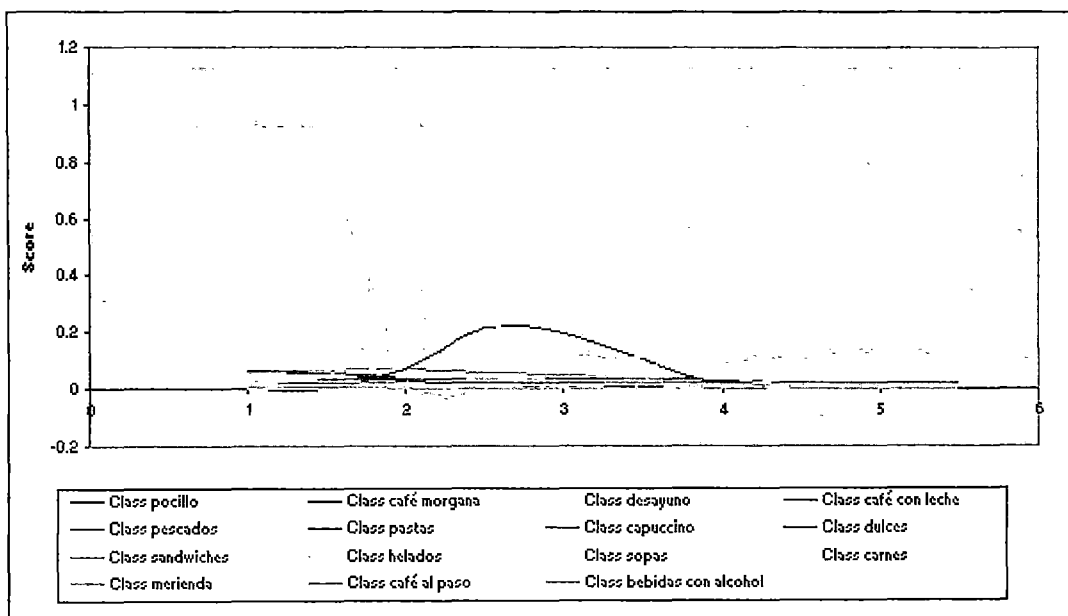


Fig. 5.5: valores asociados a la variable franja

FRANJA	Class jarra	Class café morgana	Class desayuno	Class café con leche	Class pescados	Class pastas	Class capuccino
1	0.059511598	0.0339513	0.951098	0.0569461	0.00024114	0.000449	0.021096
1.5	0.052976465	0.03528728	0.795007	0.0613405	0.00131631	0.000629	0.021058
2	0.031413117	0.03696887	0.032552	0.0688624	0.07281716	0.001549	0.018034
2.5	0.015110636	0.03364783	0.000965	0.0564714	0.21815833	0.002704	0.012832
3	0.021038516	0.03485993	0.00165	0.0482658	0.19833965	0.004118	0.018372
3.5	0.022748135	0.03370982	0.002008	0.0385169	0.10209295	0.00569	0.021261
4	0.017875332	0.02764046	0.001168	0.0186797	0.00736087	0.013394	0.022304
4.5	0.010249824	0.02259979	0.000295	0.0108941	0.00130271	0.022731	0.016993
5	0.009194919	0.02187723	0.000224	0.0101077	0.00108261	0.024578	0.01591
5.5	0.009111697	0.02181753	0.000219	0.0100437	0.00106678	0.02475	0.015822

Class dulces	Class sandwiches	Class helados	Class sopas	Class carnes	Class merienda	Class café al paso	Class bebidas con alcohol
0.063464	0.019215	0.032605	0.02105	0.008856	8.69E-08	0.005905	0.002718
0.05391	0.011885	0.042296	0.04725	0.011879	4.25E-07	0.005636	0.004548
0.033709	0.002351	0.103955	0.232707	0.018045	3.8E-05	0.003063	0.018461
0.023837	0.000588	0.220208	0.229801	0.013086	0.001121	0.001018	0.038187
0.02388	0.00176	0.137414	0.270145	0.018005	0.004941	0.001802	0.029381
0.023787	0.003362	0.109535	0.207879	0.015267	0.018495	0.001734	0.023087
0.022124	0.008148	0.10024	0.049483	0.004501	0.497013	0.000532	0.014546
0.019342	0.006267	0.128547	0.010823	0.001565	0.900153	0.000161	0.013848
0.018822	0.005579	0.133909	0.008456	0.001356	0.914893	0.000138	0.014263
0.018777	0.005525	0.134285	0.008281	0.00134	0.915898	0.000136	0.014303

Tabla 53: Listado de Resultados de la variable franja

Las gráficas de las clases en una visión tridimensional, nos muestra su comportamiento:

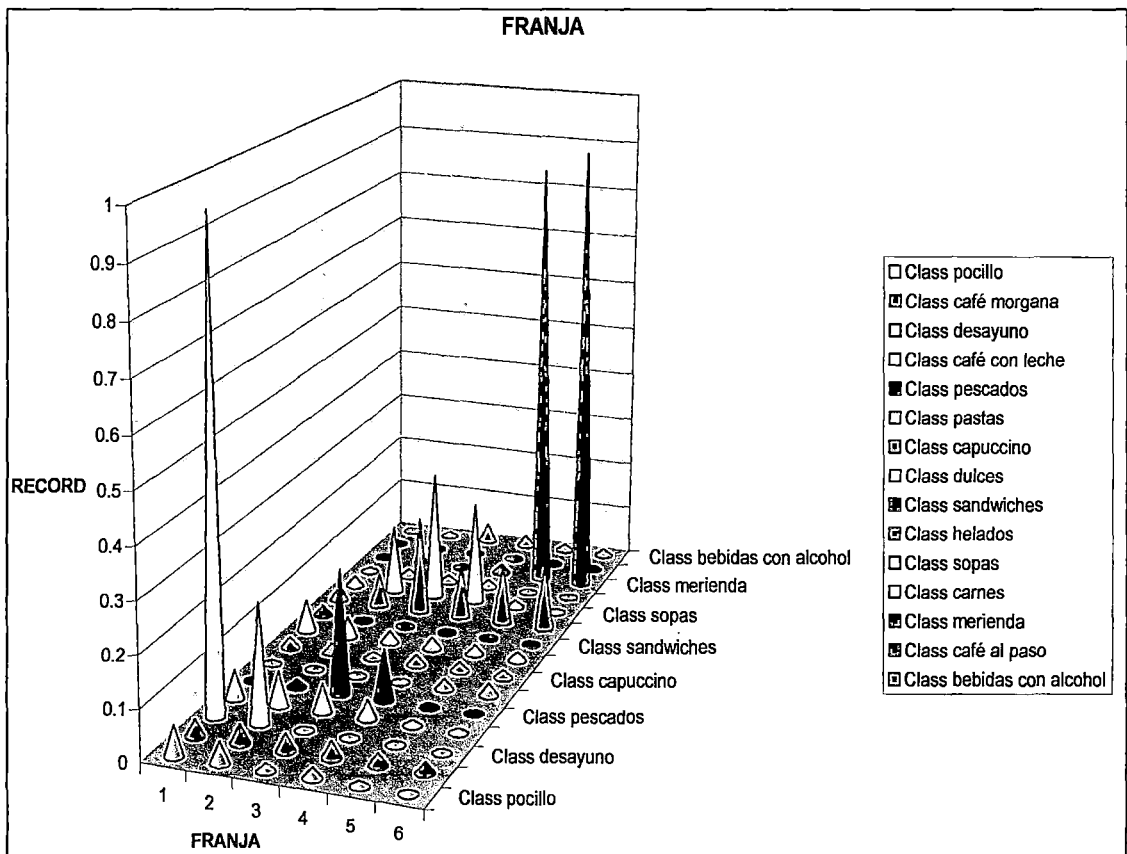


Fig. 5.6 Resultados del análisis de FRANJA

En la figura 5.6 adjunta se muestran por ejemplo, como en las horas de la mañana la clase desayuno tiene mayor consumo, en contraste con la clase merienda que tiene mayor predominio en los horarios de la noche y trasnoche.

La clase helados tiene un consumo mayor en el medio día y la tarde.

Variable CANTIDAD:

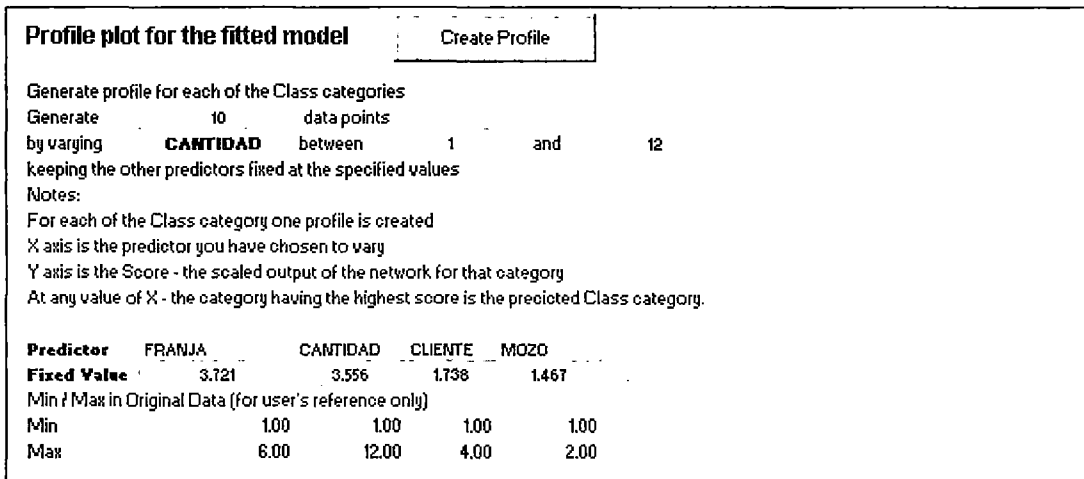


Fig. 5.7 Interfaz grafica del NNclass (Profile) Variable Cantidad

Implementada los rangos y las particiones, se activa el botón Profile, y salen las curvas y las tablas correspondientes:

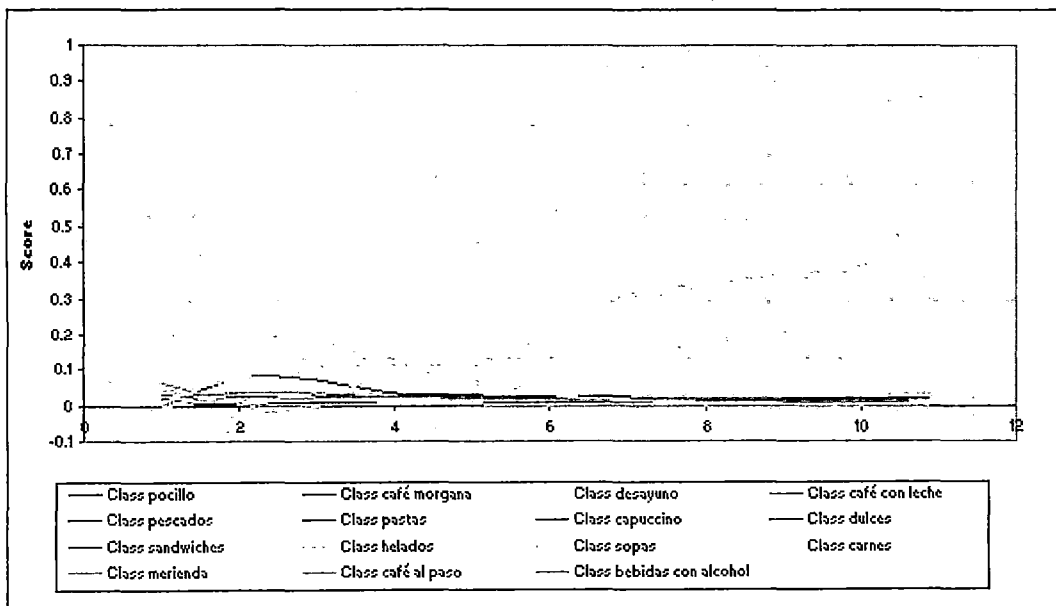


Fig. 5.8: Valores de resultados de la variable Cantidad.

En la tabla 5.4, se muestran los resultados de la variable cantidad:

CANTIDAD	Class jarra	Class café morgana	Class desayuno	Class café con leche	Class pescados	Class pastas	Class capuccino
1	0.017775704	0.02359605	0.222275	0.0067396	3.256E-05	0.002785	0.02446
2.099999905	0.025439475	0.03402514	0.003124	0.0352511	0.08267914	0.005916	0.024091
3.200000048	0.022431556	0.03264821	0.001959	0.0337467	0.06395724	0.006755	0.021972
4.300000191	0.020217033	0.03036136	0.001534	0.0263754	0.0255072	0.009032	0.021928
5.400000095	0.014605398	0.02696058	0.0007	0.0194589	0.00808158	0.011732	0.018747
6.5	0.007402669	0.02333287	0.000133	0.0163916	0.00407271	0.010945	0.011604
7.599999905	0.005301245	0.02167515	6.01E-05	0.0148367	0.00275669	0.010685	0.00921
8.699999809	0.004761136	0.02079479	4.64E-05	0.0131119	0.00180765	0.011877	0.00885
9.800000191	0.004229511	0.02006997	3.37E-05	0.011984	0.00137244	0.012755	0.008352
10.89999962	0.004055256	0.02114079	2.9E-05	0.0160862	0.00377756	0.008929	0.007265

Class dulces	Class sandwiches	Class helados	Class sopas	Class carnes	Class merienda	Class café al paso	Class bebidas con alcohol
0.029582	0.061792	0.009104	0.000336	0.035771	0.000611	0.026173	0.001019
0.02411	0.005251	0.082959	0.186332	0.019633	0.020214	0.002629	0.018699
0.023601	0.004256	0.104251	0.167168	0.012449	0.038753	0.001452	0.020822
0.022978	0.005476	0.105131	0.101819	0.007784	0.138818	0.000899	0.018059
0.021496	0.004688	0.133432	0.044394	0.003759	0.435248	0.000371	0.016949
0.019092	0.001346	0.259404	0.017125	0.001596	0.587592	9.95E-05	0.021739
0.018011	0.000753	0.339377	0.010262	0.001048	0.665562	5.28E-05	0.024093
0.017541	0.000749	0.3535	0.007414	0.000839	0.787702	4.05E-05	0.023069
0.017031	0.000679	0.376829	0.005739	0.000694	0.857179	3.14E-05	0.023078
0.01722	0.000342	0.438577	0.009575	0.000888	0.55288	3.54E-05	0.030798

Tabla 5.4: Listado de Resultados de la variable cantidad

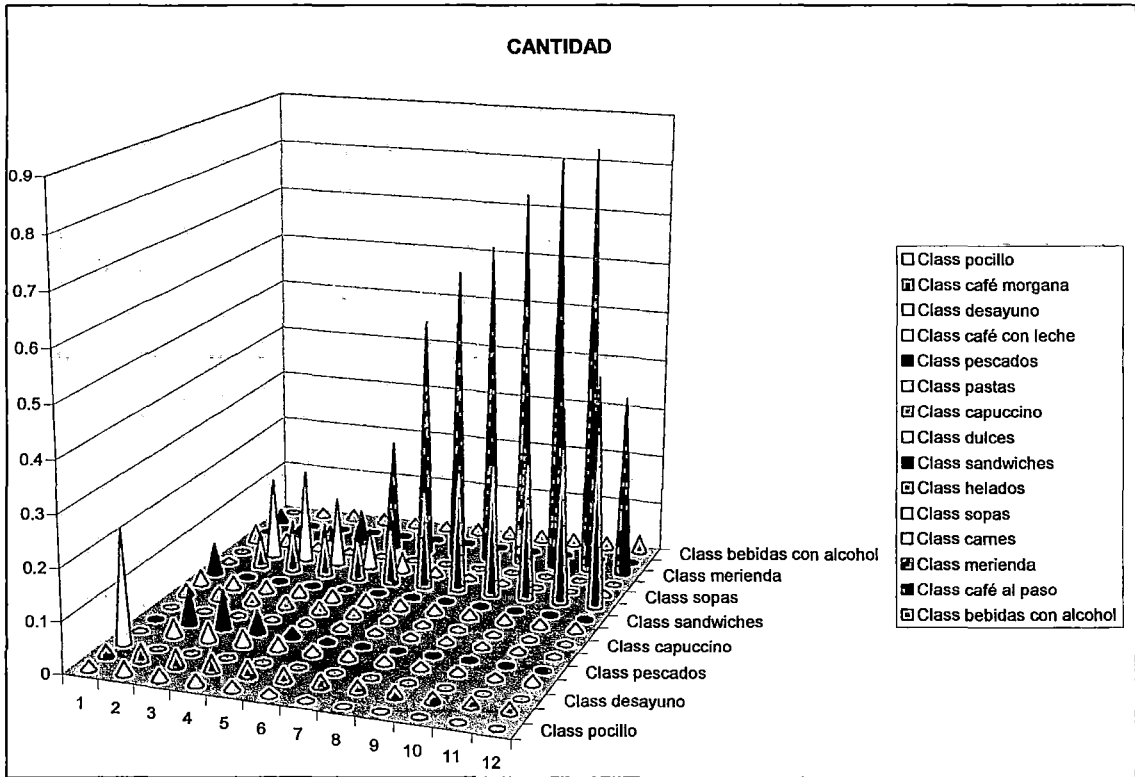


Fig. 5.9: Representación gráfica de Cantidad.

La figura 5.9 muestra a la clase merienda como la predominante en cantidad, en el rango de 10 a 11. Le sigue la clase helados con predominio en los rangos 11 a 12. Los desayunos tienen una ligera predominancia en pedidos unitarios

Variable MOZO

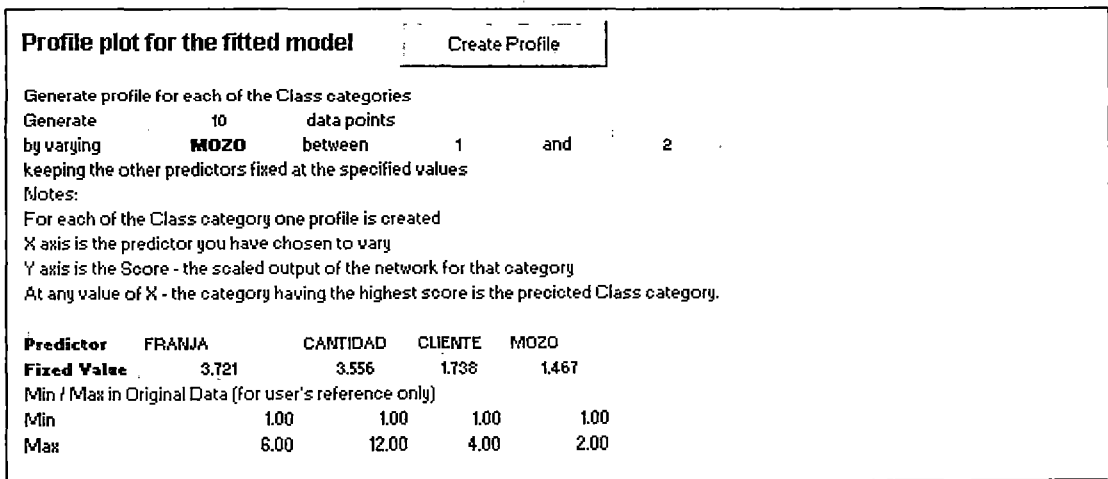


Fig. 5.10: Cuadro de análisis de la variable Mozo

Elegido los parámetros se obtienen los perfiles (fig. 5.11, 5.12) y la tabla 5.5

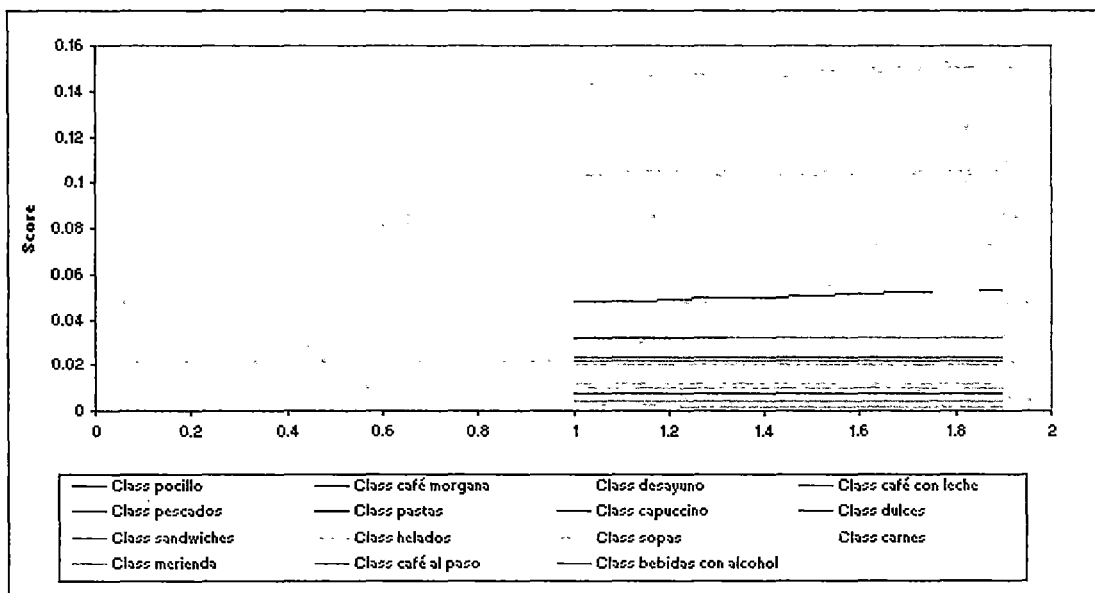


Fig. 5.11: Valores de resultados de variable Mozo.

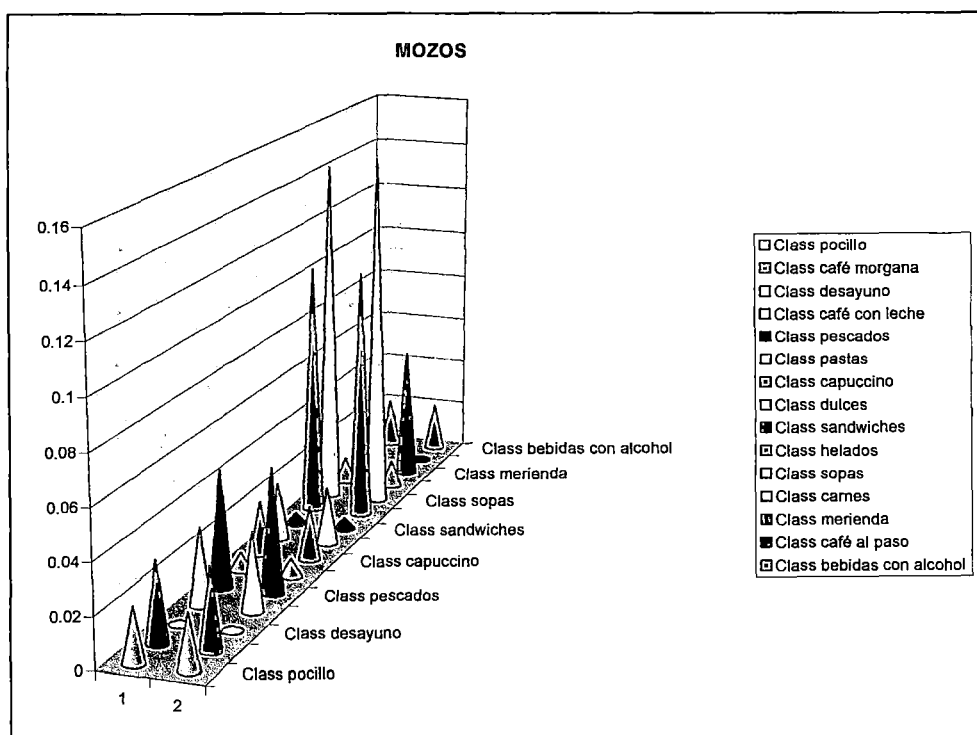


Fig.5.12: Representación gráfica de la variable Mozo

MOZO	Class jarra	Class café morgana	Class desayuno	Class café con leche	Class pescados	Class pastas	Class capuccino
1	0.021729826	0.0318969	0.001813	0.031214	0.04772627	0.007437	0.021971
1.100000024	0.021765476	0.031927	0.00182	0.0313009	0.04825236	0.007411	0.02198
1.200000048	0.021802014	0.0319588	0.001828	0.0313959	0.04882245	0.007384	0.021989
1.299999952	0.021837423	0.03199077	0.001835	0.0314938	0.0494107	0.007357	0.021996
1.399999976	0.021871409	0.03202265	0.001842	0.0315935	0.05001231	0.007329	0.022001
1.5	0.021904022	0.03205444	0.001849	0.031695	0.05062695	0.0073	0.022004
1.600000024	0.021935379	0.03208618	0.001855	0.0317983	0.0512554	0.007271	0.022006
1.700000048	0.021965607	0.03211795	0.001861	0.0319036	0.05189899	0.007242	0.022007
1.799999952	0.021994837	0.03214983	0.001867	0.032011	0.05255957	0.007213	0.022006
1.899999976	0.0220232	0.03218195	0.001873	0.0321211	0.05323966	0.007183	0.022004

Class dulces	Class sandwiches	Class helados	Class sopas	Class carnes	Class merienda	Class café al paso	Class bebidas con alcohol
0.023412	0.004623	0.104691	0.143715	0.010638	0.059651	0.001236	0.019923
0.02342	0.004615	0.104592	0.14456	0.01071	0.058733	0.001245	0.019946
0.023429	0.004605	0.104507	0.145474	0.010784	0.057763	0.001255	0.019974
0.023437	0.004594	0.104437	0.146408	0.010859	0.056788	0.001264	0.020004
0.023446	0.004582	0.104381	0.147352	0.010933	0.055817	0.001273	0.020036
0.023454	0.004568	0.104338	0.148306	0.011007	0.054851	0.001282	0.020071
0.023462	0.004553	0.104306	0.149269	0.01108	0.053889	0.001291	0.020107
0.02347	0.004538	0.104286	0.150244	0.011154	0.05293	0.0013	0.020145
0.023478	0.004521	0.104277	0.151234	0.011228	0.051973	0.001308	0.020184
0.023486	0.004503	0.104278	0.152241	0.011303	0.051015	0.001317	0.020226

Tabla 5.5: Resultados de la variable Mozo

Uno de los mozos mayor y el otro joven. Los resultados muestran que ambos han tenido un desenvolvimiento parejo, respecto de todas las clases de productos.

Variable CLIENTE

Se tienen 4 tipos de clientes definidos con los números 1, 2, 3 y 4
Elegido los parámetros se obtienen los siguientes perfiles y tablas

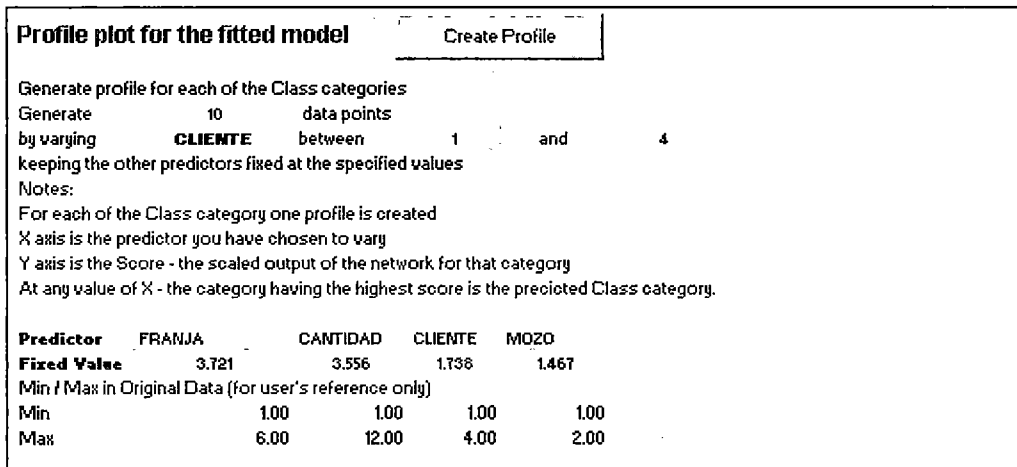


Fig. 5.13: Cuadro de análisis de la variable Cliente

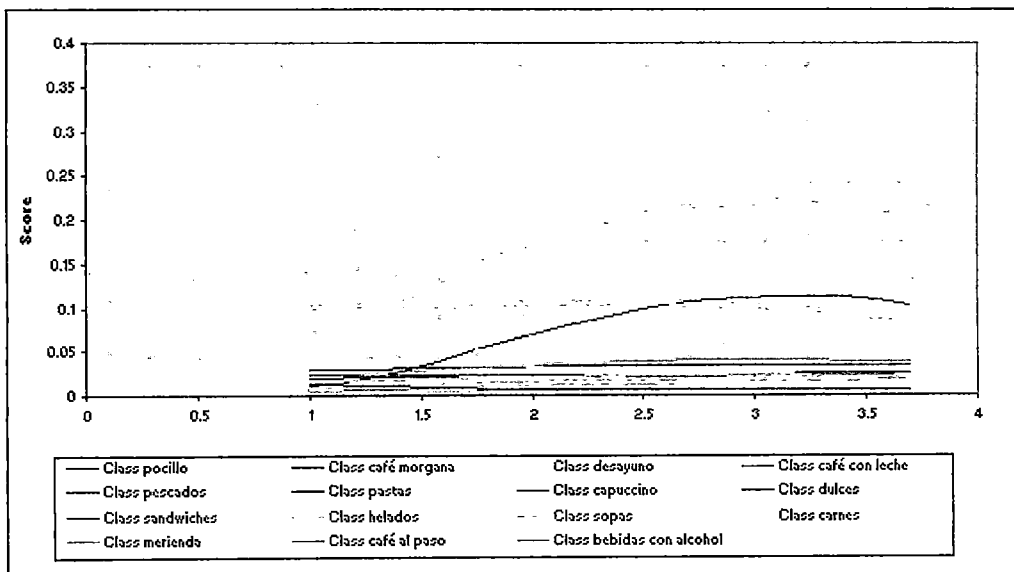
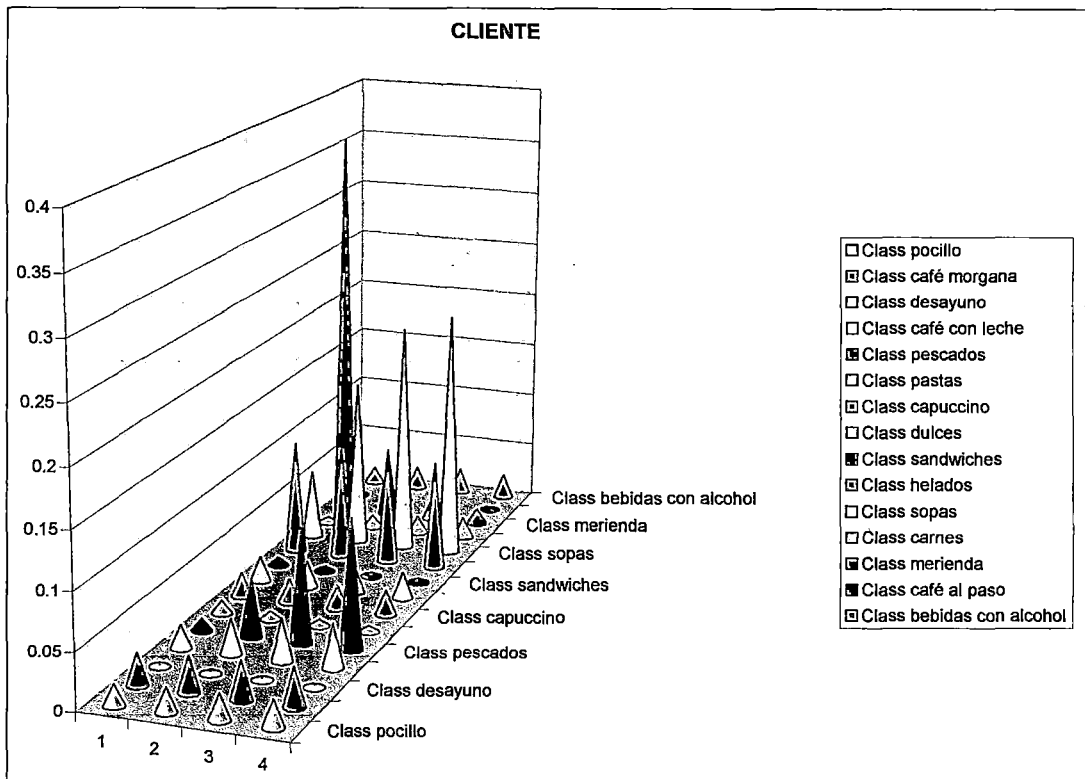


Fig. 5.14: Resultado de análisis de la variable Cliente

CLIENTE	Class jarra	Class café morgana	Class desayuno	Class café con leche	Class pescados	Class pastas	Class capuccino
1	0.018587738	0.02846641	0.001274	0.0208298	0.01083237	0.011826	0.022193
1.299999952	0.020279717	0.03011122	0.001555	0.0252982	0.02204924	0.009529	0.022297
1.600000024	0.021485596	0.03151692	0.00177	0.0298113	0.04037973	0.007863	0.02211
1.899999976	0.022278512	0.03256368	0.001922	0.0335604	0.06249905	0.006801	0.021889
2.200000048	0.022800724	0.03327822	0.002032	0.0362534	0.0830579	0.006163	0.021745
2.5	0.023188675	0.03375024	0.002126	0.0379924	0.09886246	0.005788	0.021713
2.799999952	0.023560916	0.0340697	0.002233	0.0389772	0.10926137	0.005571	0.021822
3.099999905	0.024029828	0.03430507	0.002393	0.0393235	0.11445506	0.005452	0.022128
3.400000095	0.024689526	0.03448876	0.002672	0.0389471	0.11376724	0.005395	0.022727
3.700000048	0.025495128	0.03456652	0.003181	0.0373989	0.10378501	0.005368	0.023709

Class dulces	Class sandwiches	Class helados	Class sopas	Class carnes	Class merienda	Class café al paso	Class bebidas con alcohol
0.022401	0.007199	0.102091	0.062404	0.005319	0.36128	0.000623	0.015553
0.022937	0.005996	0.102137	0.094976	0.007443	0.168002	0.000874	0.017387
0.023322	0.00494	0.103638	0.131926	0.009878	0.075737	0.001154	0.019289
0.023571	0.004235	0.105038	0.164924	0.012199	0.040169	0.001417	0.020835
0.023727	0.003824	0.105698	0.189621	0.014131	0.025921	0.001637	0.021881
0.023829	0.003617	0.105403	0.206123	0.015646	0.019544	0.001817	0.022458
0.023907	0.003565	0.103866	0.216108	0.016899	0.016394	0.00198	0.022609
0.023983	0.003669	0.100478	0.220666	0.018157	0.014744	0.002171	0.022296
0.02407	0.003991	0.094256	0.218608	0.01981	0.013899	0.00246	0.021351
0.024153	0.00468	0.084116	0.203409	0.022415	0.01359	0.002978	0.01944

Tabla 5.6: Resultados de la variable Cliente



TabFig.5.15: Representación gráfica de la variable Cliente

Los empleados bancarios son los que más consumen la merienda. El consumo de helados es uniforme en los cuatro sectores. El consumo de desayuno es mínimo en los cuatro sectores, comparado con los otros productos.

5.1.3. Conclusiones del uso de la herramienta

La herramienta NNClass ayuda a explorar una base de datos y obtener resultados interesantes relativos los patrones de consumo de los clientes. También permite descubrir los productos de bajo consumo y estos estudios pueden servir para proponer mejoras, tanto de las ofertas como cambios en las estrategias de atención.

5.2 Etapa N° 2 Estudio del Agrupamiento

5.2.1. Aplicación de la Herramienta: NNClust

El NNClust es una herramienta de aplicación de las redes de neuronales de Kohonen – orientada al cliente. Puesto que la herramienta soporta datos

numéricos, el número de campos a utilizar se puede considerar 03 parámetros: Cantidad vendida del producto, Precio del producto y Total de venta.

Sin embargo, la estructura de datos que nos proporcionó el gerente de la empresa, no fue muy útil para el análisis de segmentación que se quería realizar, pues el clustering o segmentación se aplica a los clientes y con la información disponible se logró una segmentación.

En la Fig. 5.16 se presenta la Unidad de Interfaz gráfica de la herramienta NNClust.

Neural Network based Clustering		(Using Kohonen's Self Organizing Maps (SOM))	
Number of observations	1000	Learning parameter (should be >0 and <1)	
(Needs to be between 5 and 5,000)		Start value	0,8
		End value	0,1
		Decay	Exponential
Number of Variables	3	Sigma for the Gaussian neighborhood	
(Needs to be between 3 and 50)		as % of map width (should be > 0% and < 100%)	
Enter <i>n</i> , where <i>n</i> -Square = # neurons in the map	3	Start value	50,0%
(<i>n</i> needs to be between 2 and 10)		End value	1,0%
		Decay	Exponential
[Note: By entering <i>n</i> you are specifying that the maximum number of clusters will be at most <i>n</i> -square. e.g. if you enter <i>n</i> = 4, you will get less than or equal to 16 clusters]			
Number of training cycles	50		
(Needs to be between 1 and 1000)			
		Build Clusters	
Randomize the order in which inputs are presented to the map ?	No		

Fig. 5.16 Unidad de Interfaz Gráfica de NNClust

Terminado este procesamiento, se utilizaron las tres variables numéricas siguientes: la cantidad de productos en cada factura, el total de dicha factura y el precio promedio por artículo. Se utilizaron 1000 datos de la base de datos, 3 neuronas, lo cual permitió una diferenciación máxima de 09 clusters, 50 ciclos de entrenamiento y los parámetros de aprendizaje considerando los rangos establecidos en la herramienta. Todo esto permitió no extender exageradamente el tiempo de ejecución.

La primera prueba se hizo utilizando los valores indicados y los pesos

proporcionados por el programa. Esto generó un conjunto de resultados que se muestran en los clusters obtenidos como se muestra en la Fig. 5.17, que discrimina dos grupos de clientes densos, un tercero menos denso y un cuarto que es muy selectivo. A continuación se realizó una segunda compilación utilizando los resultados de la compilación anterior obteniendo resultados similares a los resultados de la primera compilación como se muestra más adelante.

5.2.2. Resultados Obtenidos

5.2.2.1 Resultados de la Primera Compilación

Al realizar la primera compilación del programa se obtuvo como resultado dos clusters principales, una tercera menos concentrada y una cuarta de "outliers" (valores atípicos). La figura 5.17. Muestra los resultados de esta primera compilación.

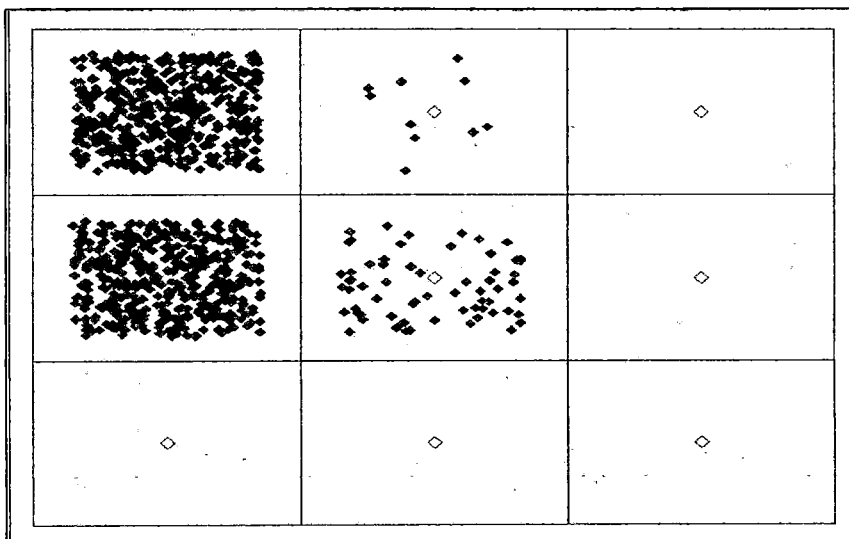


Figura 5.17. Gráfico de primera compilación del NNClust.

En la Fig. 5.17 se puede notar que existe una gran concentración de datos en dos clusters en particular. Para hacer el análisis y descripción de los clusters utilizaremos los resultados obtenidos haciendo click en la pestaña Output, que se observan en la tabla 5.7

	En	Cluster			
	General	1	2	3	4
Columna en el mapa		1	1	2	2
Fila en el mapa		1	2	2	1
Tamaño del Cluster	1000	451	474	65	10
TOTAL de Factura	20.0	10.0	30.7	12.3	16.1
CANTIDAD de productos	5.1	2.8	7.8	2.0	3.2
PRECIO por producto	4.0	3.6	4.1	6.0	5.1

Tabla 5.7. Resultados de la primera compilación del NNClust.

Los clusters más representativos, en cuanto a la cantidad de datos que acumulan, son el 1 (46%) y el 2 (47.4%), que se encuentran en la columna 1 y fila 1, y fila 2 respectivamente. Los clusters menos representativos se encuentran en la columna 2 y s filas 1 y 2 respectivamente. En cuanto a la facturación total predomina el cluster 2 con el 30.7%. El cluster 1 contiene el 10.0% de la facturación total, el cluster 3 representa el 12.3% de la facturación total. Y el cluster 4 curiosamente tiene el 16.1% de la facturación. La suma no es el 100%, pues en el agrupamiento quedan una parte fuera de los clusters.

Se observa que los clusters de la primera columna tienen mayor población, concentrando un porcentaje mayor de datos. En el cluster 1 el consumo es bajo en cantidad y en precio. Probablemente sean los clientes que toman solo un café. Los otros dos clusters de columna 2 tienen una población menor, con un consumo de productos de mayor precio y cantidades bajas, pero con una facturación superior al cluster 1.

La gráfica 5.18 ayuda a entender lo descrito hasta este momento, respecto de los cuatro clusters. Debe también prestarse atención a los clusters de menor tamaño, porque son de alta rentabilidad, en consecuencia podrían incentivarse su venta asociadas a otros productos agrupados en los clusters de mayor tamaño.

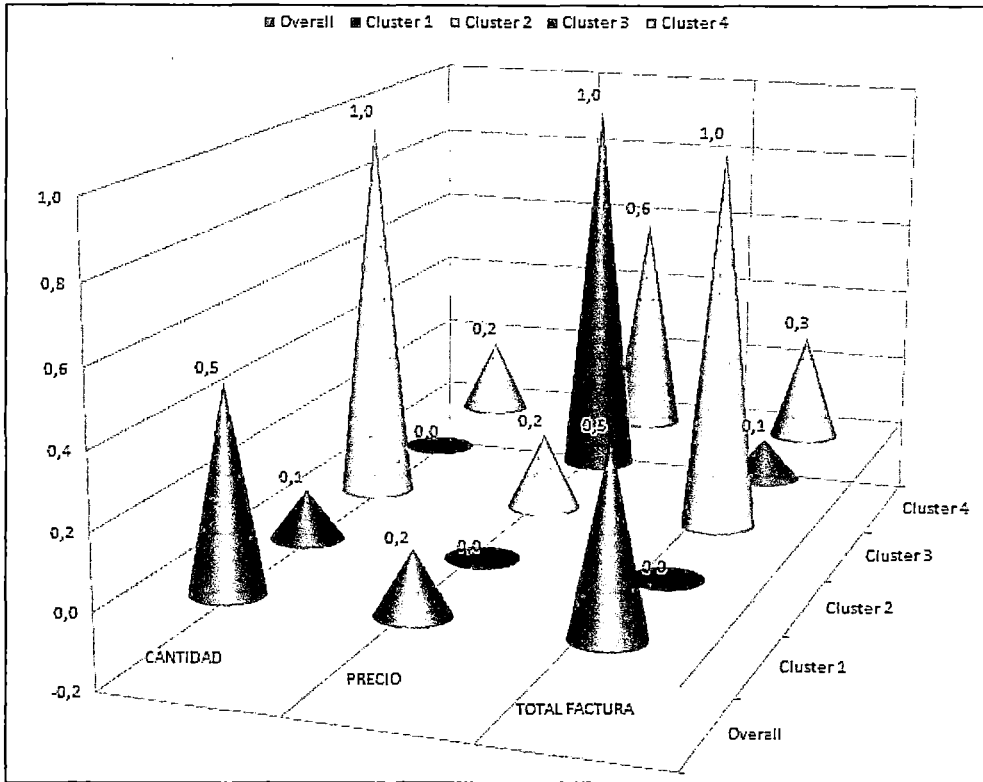


Fig. 5.18 Valores finales de parámetro de aprendizaje

5.2.2.2 Resultados de la Segunda Compilación

En la figura 5.19 al realizar la segunda compilación del programa se obtuvo como resultado dos clusters principales y una serie de "outliers" ubicados en dos sectores secundarios. La tabla 5.8. Muestra los resultados de esta segunda compilación.

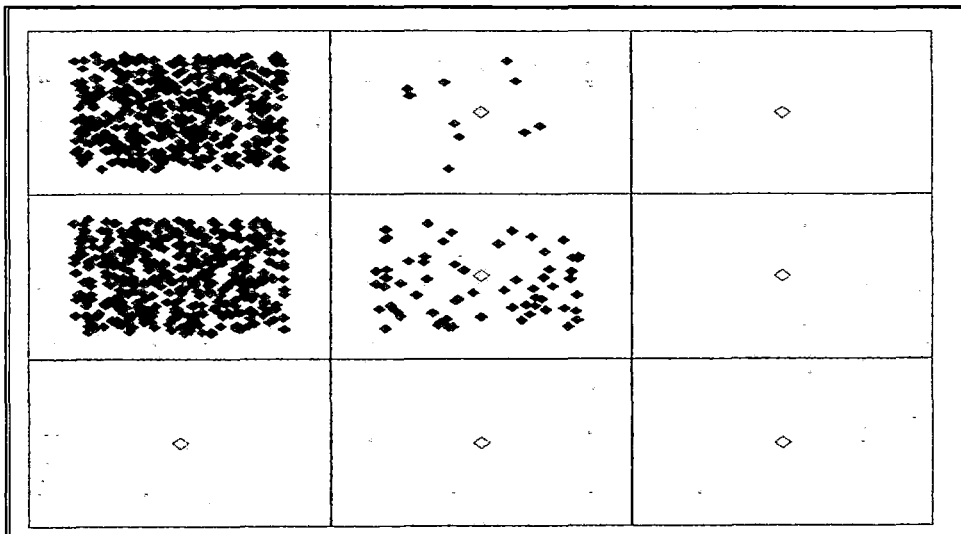


Fig. 5.19 Gráfico de segunda compilación del NNClust

	En	Cluster			
	General	1	2	3	4
Columna en el mapa		1	1	1	2
Fila en el mapa		1	2	3	2
Tamaño del Cluster	1000	451	474	65	10
TOTAL de Factura	20.0	10.0	30.7	12.3	16.1
CANTIDAD de productos	5.1	2.8	7.8	2.0	3.2
PRECIO por producto	4.0	3.6	4.1	6.0	5.1

Tabla 5.8. Resultados de la segunda compilación del NNClust.

La segunda compilación tiene un poder de discriminación muy similar al de la primera, pues el número de clusters permite visualizar los mismos grupos representativos

El cluster 1 concentra un gran porcentaje de clientes y sigue representando a los consumidores que proporcionan baja facturación. En éstas existe toda una gama de facturas de consumos medios con distintas combinaciones de cantidad de productos y promedio de precios.

Es evidente que el nivel de aprendizaje del programa es alto pues en la segunda compilación se obtienen resultados muy similares a la primera. Si bien los resultados de esta compilación son válidos, es conveniente realizar otra compilación para verificar si se logra una mayor segmentación de los clientes.

5.2.2.3 Resultados de la Tercera Compilación

En esta compilación se obtuvieron 6 cluster, de los cuales solo uno es el más representativo, respecto del tamaño del cluster. La figura 5.20 y la tabla 5.9 muestran los resultados.

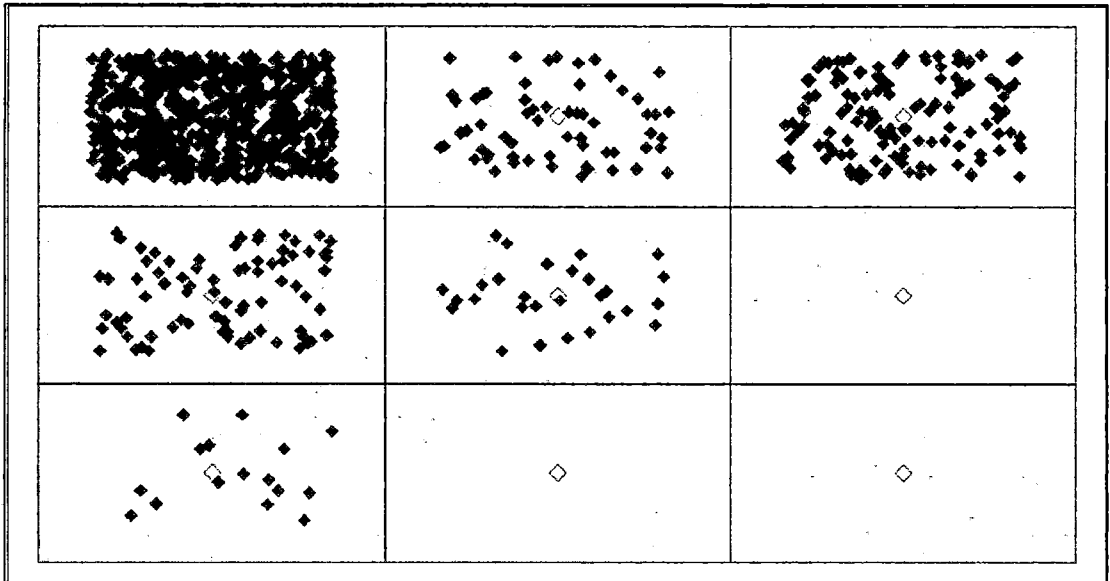


Figura 5.20. Gráfico de la tercera compilación del NNClust.

	General	Cluster					
		1	2	3	4	5	6
Columna en el mapa		1	1	1	2	2	3
Fila en el mapa		1	2	3	1	2	1
Tamaño del Cluster	1000	677	78	16	63	28	138
TOTAL de la Factura	20.0	17.0	18.7	19.5	37.2	21.4	19.5
CANTIDAD de ítems	4.0	3.7	4.0	4.0	5.3	3.9	4.0
PRECIO por ítem	5.1	4.8	5.0	5.0	6.9	5.8	5.1

Tabla 5.9. Resultados de la tercera compilación del NNClust.

Por las razones expuestas, es difícil ver si la tercera compilación realmente aporta al objetivo del trabajo. Se hace necesario analizar detalladamente los nuevos clusters que varían entre la segunda y tercera compilación.

	Segunda Compilación		Tercera Compilación			
	1	2	1	2	4	6
Columna en el mapa	1	1	1	1	2	3
Fila en el mapa	1	2	1	2	1	1
Tamaño del Cluster	451	474	677	78	63	138
TOTAL de la Factura	10.0	30.7	17.0	18.7	37.2	19.5
CANTIDAD de ítems	5.1	2.8	3.7	4.0	5.3	4.0
PRECIO por ítem	4.0	3.6	4.8	5.0	6.9	5.1

Tabla 5.10. Comparación de clusters entre segunda y tercera compilación.

Se puede apreciar en la tabla que los clusters restantes de la tercera compilación son diferentes entre sí, y a su vez difieren de los de la segunda compilación. Estas diferencias se parecían sobre todo en los campos TOTAL y PROMEDIO. Por otro lado, gran parte de los nuevos clusters supera los 100 datos, lo cual les otorga la representatividad necesaria. De este modo se puede concluir que la tercera compilación es útil para el análisis ya que proporciona nueva información sobre los clientes de mayor consumo.

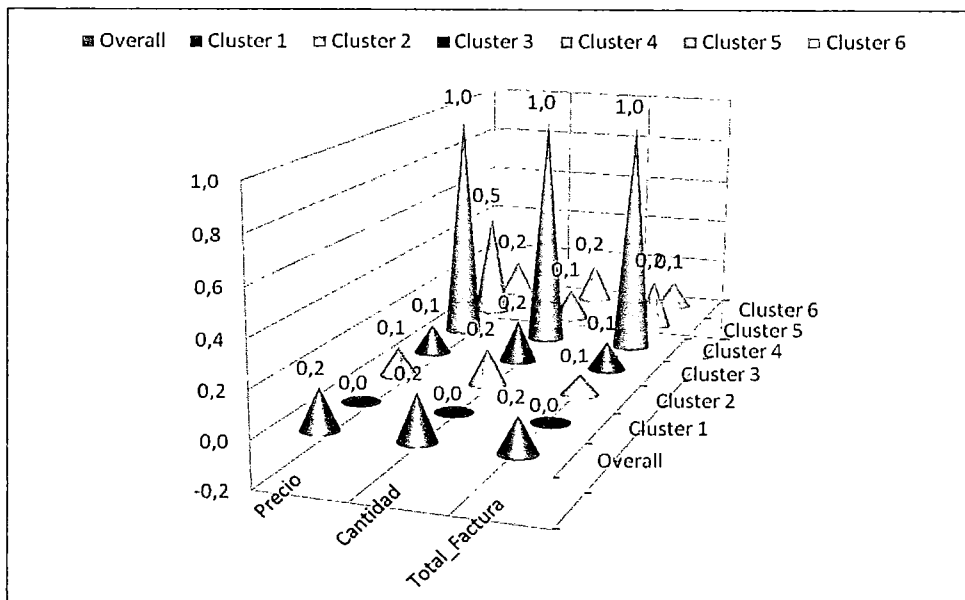


Fig. 5.21 Valores finales de parámetro de aprendizaje

5.2.2.4 Análisis de los resultados

Teniendo en cuenta el nivel de representatividad de los clúster obtenidos en la tercera compilación, fueron seleccionados para realizar el análisis de la clientela del local. La tabla 5.10 muestra los resultados de dicha compilación.

Representan el porcentaje de la población de cada cluster respecto al total de los datos, la facturación total de cada clúster en el periodo de análisis, expresada como el consumo promedio de cada factura por la población del cluster, y por último el porcentaje de facturación.

5.2.2.5 Conclusiones del uso de la herramienta

El análisis de los clúster tiene como objetivo segmentar a los clientes para analizar sus características y tomar decisiones acertadas y oportunas en beneficio del negocio. Mediante dichos resultados se puede evaluar el proceso interno e introducir promociones para generar mayor rentabilidad.

Si el análisis se hace sobre un grupo pequeño de clientes es posible obtener los mismos o mejores resultados que tratando de abarcar a la totalidad de clientes, con un costo notablemente menor. Esto muestra la importancia de hacer un análisis exhaustivo de los datos existentes.

Otro aspecto que se debe tener en cuenta al analizar los clústers es el factor humano. No se pueden estudiar los segmentos como meros números, sino que se deben tener en cuenta las características de los clientes, de modo que las nuevas estrategias de posicionamiento, deberán tener en cuenta la parte humana.

CONCLUSIONES Y RECOMENDACIONES

1. CONCLUSIONES:

- El modelo planteado basado en Redes Neuronales Backpropagation y de Kohonen, juntamente con las herramientas de libre disponibilidad donde están implementadas: NNClass y NNClust, permiten aplicar técnicas de clasificación y agrupamiento respectivamente a un negocio, para identificar patrones de consumo que ayudan a formular propuestas de mejora de la oferta y del servicio (5.1.3, 5.2.5, 5.2.6).
- La herramienta NNClust permite analizar los datos de la empresa para obtener una segmentación de los clientes y analizar sus características y hacer propuestas para el negocio a la hora de hacer cambios e introducir promociones.
- La herramienta NNClass permite analizar los datos de la empresa para obtener patrones de consumo de distintos productos en función a las franjas horarias, al día de los clientes que concurren al local. (5.2.6)
- No existe una arquitectura ideal de red neuronal para todo tipo de aplicaciones. La arquitectura más apropiada se va obteniendo a través de sucesivos entrenamientos o por ensayo y error. En las dos herramientas se tienen restricciones al número de neuronas y capas y por la lentitud en su ejecución, es suficiente una red con una o dos capas ocultas, y a lo más cinco neuronas en cada una de ellas para obtener resultados satisfactorios. (2.5.9).

- Se ha demostrado la sencillez y la utilidad del uso de las herramientas de redes neuronales para identificar patrones de consumo en un sistema de negocio utilizando las técnicas de clasificación y agrupamiento. (5.1.6, 5.2.3).
- Se ha demostrado el potencial de las redes neuronales en la aplicación a negocios pequeños o medianos, dado su costo cero. Cualquiera de estas empresas puede utilizar para mejorar su oferta y de este modo satisfacer las necesidades de sus clientes. (5.2.6).

2. RECOMENDACIONES:

- Si se desea identificar patrones de consumo en una organización pequeña o mediana para realizar cambios y hacer promociones, se recomienda utilizar técnicas de clasificación y agrupamiento y las herramientas de libre disponibilidad NNClass y NNClust.
- Si una organización desea utilizar Redes Neuronales para identificar patrones de consumo, se recomienda contar con una base de datos, adecuadamente organizada y depurada, para evitar el trabajo engorroso de depuración.
- Al aplicar RNA para identificar patrones de consumo, se recomienda conocer las reglas del negocio para seleccionar las variables relevantes que deben ser usadas como insumo de la red neuronal seleccionada. Estas reglas deben ser obtenidas a partir de la Base histórica de datos del negocio.

- Este trabajo contiene una aplicación específica de las redes neuronales a los negocios, empleando una base de datos de una cafetería. Se recomienda aplicar el modelo propuesto a otros rubros y negocios como: empresas de servicios, industria manufacturera, empresas de energía, compañías financieras, entre otras.
- Se recomienda utilizar herramientas más sofisticadas que no posean las limitaciones de las herramientas utilizadas en este trabajo (NNClust y NNClass). Así mismo sería interesante utilizar otros tipos de redes neuronales en esta clase de problemas.

GLOSARIO DE TERMINOS

Minería de Datos: es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Redes Neuronales Artificiales (RNA): son modelos que intentan reproducir el comportamiento del cerebro capaz de realizar una determinada tarea. Estos modelos poseen dispositivos elementales de proceso, las neuronas, al igual que en el modelo biológico. Las redes auto organizadas (de kohonen) son modelos orientados a la clusterización.

Función de activación: Es una función utilizada al modelar matemáticamente una red neuronal artificial. La más utilizada en la actualidad es la función logística o sigmoidea.

Perceptrón Multicapa (MLP): Primer modelo de RNA capaz de asociar procesos fue el perceptrón. El perceptrón Multicapa (MLP) es un modelo de RNA que junto a un método de aprendizaje especial. El algoritmo de retropropagación, es capaz de resolver problemas asociando patrones.

Algoritmo Backpropagation: Consta de tres fases: fase de aprendizaje o entrenamiento, fase de validación y la fase de test.

Arquitectura de una RNA: En la organización dentro de una RNA,

generalmente las redes neuronales están asociadas por capas: capa de entrada, capa(s) oculta(s) y capa de salida.

Aprendizaje no supervisado: Se suministran los inputs de entrada, pero la red no recibe ninguna información que le indique cual es la salida. Son de dos tipos: el hebbiano y el competitivo/ corporativo

Segmentación de base de datos: Cluster no jerárquico. Se compara cada registro de la base de datos con todos los segmentos creados. Se mide la distancia del registro de entrada con segmentos ya creados y se asigna el registro de entrada al segmento correspondiente.

Modelos de clasificación: En estos modelos se utiliza aprendizaje supervisado. Estos modelos suelen utilizar árboles de decisión y un conjunto de datos de entrenamiento para crear el modelo, que posteriormente se utiliza para clasificar los datos.

Herramientas de libre disponibilidad: software de distribución gratuita por internet. Ejemplo: NNClust y NNClass.

NNClust: Herramienta que permite el análisis automático de datos que utiliza los mapas auto organizados de Kohonen para el "clustering" o segmentación de datos. Son un tipo especial de RNA no supervisadas.

NNClass: Herramienta que permite el análisis automático de datos utilizando redes neuronales de retropropagación (Backpropagation)

REFERENCIAS BIBLIOGRÁFICAS

1. Acosta M. I., Balazar, H. y Zuloaga C. "Tutorial de Redes Neuronales", Universidad Tecnológica de Pereira, Colombia, 2000. Disponible en <http://ohm.utp.edu.co/neuronales/Capitulo2/Backpropagation/AntecedentesB.htm>, página web vigente al 31/05/2006.
2. Acuña Fernández, Edgar: "Análisis de Datos Estadísticos usando Minitab para Windows", Lima. 1999.
3. Britos, P., "Objetivos de Negocio y Proceso de Minería de Datos Basados en Sistemas Inteligentes", "Reportes Técnicos en Ingeniería del Software", núm. 7, págs. 26-29-2005.
4. Checkland Peter: "Metodología de Sistemas Suaves en Acción", Editorial Wiley. México. 1994.
5. Comisión Económica para América Latina: "Análisis de la Competitividad del Sector Industrial Peruano - Metodología CAN PLUS", Perspectiva Industrial N°4. Lima.2000.
6. Espinoza Haro, Pedro C. "Matemática para el Moldeamiento de Sistemas de Producción y Operaciones". S.PG FIIS –UNI Lima 2010.
7. Figueroa Palomino, Renzo, José. "Metodología y Análisis Comparativo de la Situación Productiva y Competitiva del Sector Industrial Manufacturera Mediante la Aplicación de Redes Neuronales y Estadísticas Multivariadas". Tesis para optar el grado académico de Maestro en Ciencias con Mención en Ingeniería de Sistemas. SPG FIIS – UNI, Lima 2007.
8. [Fiszelew, A. y García Martínez, R. 2002] Generación Automática de

- Redes Neuronales con Ajuste de Parámetros Basado en Algoritmos Genéticos. Revista del Instituto Tecnológico de Buenos Aires, 26: 76-101.
9. García Vega A. "La Inteligencia Artificial en los Negocios", Universidad de Veracruz, 2002.
 10. García Martínez, R.; Servente, M.; Pasquín, D.; 2003. Sistemas Inteligentes, Capítulo 1: "Aprendizaje Automático", Capítulo 2 "Redes Neuronales Artificiales"; Nueva Librería, Buenos Aires, Argentina.
 11. Gil Alborran Guillermo E. "Data mining Minería de datos y SQL". Ed. Megabytes, 2006 Lima.
 12. Gómez de Lima Manuel Sebastián. Tesis de Grado en Ingeniería Industrial. Redes Neuronales Aplicadas a los Negocios: Análisis de Un caso. Instituto Tecnológico de Buenos Aires, Argentina. 2006.
 13. Hiler José R. y Martínez Víctor J.:"Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones", RA-MA Editorial. España. 2000.
 14. Isasi Vinueda Pedro, Galván León, Inés M. Redes Neuronales un enfoque práctico. Ed Pearson Educación S.A España 2004.
 15. Jiménez Félix, Aguilar Giovanna y Kapsoli Javier: "Competitividad en la Industria Manufacturera Peruana", Documento de Trabajo N° 148, 1998.
 16. Jiménez Félix, Aguilar Giovanna y Kapsoli Javier: "Competitividad, rendimientos crecientes y comercio intra-industrial en la Manufactura peruana", Documento de Trabajo N° 153, 1998.
 17. Joya C, Gonzales, Atencio R. Miguel, Ochoia R. Alberto, Allende A. Siza; "Optimización Inteligente Técnicas de Inteligencia Computacional para Optimización", servicio de Publicaciones e Intercambio cultural de la Universidad de Málaga. Málaga, 2004.
 18. [Kohonen, T. (1988)] Self-Organizing Maps Springer Series in Information Sciences Vol. 30, Springer, Berlin, Heidelberg, NY(pp 236)
 19. Koskivaara Eija: "Artificial Neural Networks in Auditing: State of the Art", TUCS Technical Report No 509. Finlandia.2003

20. Mallo González, Carlos: "Un modelo de Red Neuronal Artificial aplicado en la predicción de la demanda horaria de energía eléctrica", España.2003.
21. Martin del Brio, B; Serrano, C: "Self-Organizing Neural Networks: The Financial State of Spanish Companies". Neural Networks in the Capital Markets, Ed Refenes, John Wiley & Sons, Capitulo XXIII. (1995)
22. Mat Junoh, Mohd: "Predicting GDP growth in Malaysia using knowledge-based economy indicators: a comparison between neural network and econometric approaches", Sunway College Journal, Vol 1. pp 39-50. Malaysia.2004.
23. Montaña Moreno, Juan: "Redes Neuronales Artificiales aplicados al análisis de datos", Tesis Doctoral, Universidad de las Islas Baleares, Palma de Mallorca. 2002.
24. Angshuman Saha; Tesis Doctoral Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models; Departamento de Estadística; Universidad de Washington; Agosto 1998. <http://www.geocities.com/adotsaha/NN/SOMinExcel.html>. Agosto 1998.
25. Noket, S., Britos, P. y García Martínez, R.: "Pronóstico de ventas basada en la comparación de la técnicas de redes neuronales y método estadístico", Departamento de Ingeniería Industrial, Argentina.2003.
26. Palmer, A., Montaña, J.J. y Jiménez, R.: "Tutorial sobre redes neuronales Artificiales - el perceptrón multicapa", Revista electrónica N°5, España.2001.
27. Pérez Cesar, Santen Daniel, "Data Mining, Soluciones con Enterprise Minna" Ed Alfa y Omega Grupo Editor S.A México 2006.
28. Pitarque Alfonso, Roy Juan Francisco y Ruiz Juan Carlos: "Redes Neurales versus Modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación", Valencia. España. 1998.
29. Rapaz Diego, "Aplicación de Minería de Datos para Determinar la

- eficacia de la Braquiterapia en el tratamiento de cáncer de Próstata”. Tesis de Grado en Ingeniería Industrial. ITBA, Buenos Aires, 2008.
30. [Rosenblatt, F. (1958)] The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
 31. [Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. (1986)] Learning internal representations by back-propagating errors in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Eds. Cambridge, MA: MIT Press, vol. 1, p. 318-362.
 32. [Saha, A. (1998) Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models] Tesis Doctoral. Departamento de Estadística;
 33. Sanchez Comperas E., Alania Garcia Alma Yolanda. *Redes Neuronales*. Ed. Pearson Prentice – Hall. España 2006.
 34. Sandivar Rosas, Juana. “Pronostico de la Demanda de Postulantes de un Centro Superior Universitario, Basado en Redes Neuronales”. Tesis para optar el grado académico de Maestro en Ciencias con Mención en Ingeniería de Sistemas. SPG FIIS – UNI, Lima 2008.
 35. Gómez Rodríguez, Gladis Gloria “Sistemas de Soporte a las Decisiones con Data Mining. Caso: Gestión de Deuda en la Administración Tributaria”. Tesis para optar el grado académico de Maestro en Ingeniería de Sistemas. Escuela de Postgrado de la Universidad de Lima. Junio 2007.
 36. [Stephan, C; Cammann, H; Jung, K. 2005] Artificial Neural networks: Has the time come for their use in prostate cancer patients. *Nature*, 2: 262-263.
 37. Torres, Zorrilla Jorge: “Cluster de la Industria en el Perú”, Documento de Trabajo N° 228, Lima 2003.
 38. Vilcapoma Ecurra, Edgar S; “Identificación de Patrones de Evasión en el Sistema de Administración Tributaria utilizando tecnología Data

Mining”. Tesis para optar el grado académico de: Maestro en Ciencias con mención en Ingeniería de Sistemas. SPG FIIS - UNI, Lima 2004

ANEXOS

ANEXO 1

Requerimiento de Información

Sres. de la empresa "Café Morgana"

La base de datos corresponde a las ventas de un local de "Café Morgana", cada fila en la base de datos corresponde a un ítem dentro de un recibo de atención al cliente.

El lapso de tiempo de la información requerida de forma óptima sería de un año de ventas para un local en particular, en todo caso podría ser por meses en particulares.

Campo	Descripción
Artículo	Contiene un código para cada tipo de artículo vendido en cada atención.
Descripción de Artículo	Nombre o descripción del artículo que se vende. Ej. Café Morgana
Cantidad	La cantidad comprada del artículo en cuestión.
Fecha	Fecha de la Venta. Si es texto formato DD/MM/AAAA
Hora	Hora de la Venta. Si es texto formato HH:MI
N° Recibo	Es el número que identifica al recibo de la atención. Por recibo entendemos cualquier documento de cobro que se emita. Ej. Boleta, Factura, etc.
N° de renglón	Es un contador de líneas dentro del recibo de atención. En caso que se haya comprado dos artículos diferentes en un solo recibo de atención se tendrá en este campo el valor 1 mientras que el segundo tendrá un 2.
Precio	Precio del artículo en moneda nacional.
Total de la Factura	Importe que contabiliza el total a abonar por la atención recibida. Importe en moneda nacional.
Código de Rubro	Los artículos se agrupan en rubros diferentes.
Descripción de rubro	Cada código de rubro tiene una descripción en palabras. Ej. Ensaladas
Descripción del artículo	Si la venta es por una oferta o promoción, describirlo brevemente.

Estructura de la información requerida:

El formato debidamente llenado deberá ser remitido a más tardar el 28 de noviembre de 2010 a través del correo electrónico jangulope@hotmail.com.

Finalmente, le indicamos que cualquier consulta sobre el particular deberá ser canalizada a través del correo antes indicado.

Lima, 20 de Diciembre de 2010

Anexo 2 Características de las PYMES Industriales

Para el análisis a nivel sectorial se ha considerado el período 1994-1999. El año 1992 no será tomado en cuenta debido a la metodología utilizada para la elaboración de la Encuesta Económica Anual de Estadística Manufacturera. Cuando se traten de datos agregados a nivel industrial, la comparación si es factible.

Con este análisis vamos a poder observar la importancia y el grado de participación que tienen las PYMES a nivel industrial en las diferentes variables en estudio, así como en el contexto de la economía nacional.

1. Análisis General de la Información

a. Definiciones de PYME

La definición de pequeña y mediana empresa (PYME) en el Perú varía de acuerdo al criterio de medición. Al respecto, el Ministerio de Industria, Turismo, Integración y Negociaciones Comerciales Internacionales (MITINCI) desarrolló esfuerzos para unificar conceptos e información.

El Decreto Legislativo N° 705, promulgado el 15 de Noviembre de 1991 hace referencia únicamente a micro y pequeñas empresas. Se considera microempresa a la unidad económica que cuenta con no más de 10 trabajadores y cuyas ventas anuales no son mayores a 12 unidades impositivas tributarias (UIT). La pequeña empresa es aquella que emplea a no más de 20 trabajadores y cuyas ventas anuales no superan las 25 UIT.

CUADRO N° 1 Definiciones de tamaño de empresa

Tamaño	Número de trabajadores	Ventas anuales.
Micro empresa ^a	No exceda a 10	Hasta 12 UIT
Pequeña empresa ^a	No exceda a 20	Hasta 25 UIT
Mediana empresa ^b	Entre 21 y 200	
Gran empresa ^b	Más de 200	

^a Basada en el Decreto Legislativo N° 705

^b Basada en la definición más utilizada

A nivel de América Latina los criterios para definir las pequeñas y medianas empresas también varían, ya que en algunos países la clasificación se hace en función a las ventas y en otros al empleo. el cuadro N° 2 presentaremos a modo de ilustración algunas definiciones que se manejan en algunos países de América Latina.

CUADRO N° 2

Definiciones de la Micro, Pequeña y Mediana Empresa a Nivel Latinoamericano.

País	Criterio	Micro	Pequeña	Mediana
Argentina ¹	Ventas	Hasta 0.9	Hasta 5.4	Hasta 43.2
Chile ²	Ventas	Hasta 2,400	Hasta 25,000	Hasta 100,000
Colombia ³	Empleo	Hasta 10	Hasta 50	Hasta 200
Venezuela ⁴	Empleo	Hasta 10	Hasta 50	Hasta 100
Bolivia ⁵	Empleo	Hasta 10	Hasta 19	Hasta 49
Costa Rica ⁶	Empleo	Hasta 10	Hasta 30	Hasta 100

Fuente: Panorama de las Micro, Pequeñas y Medianas Empresas

(MiPyMEs) en varios Países de América Latina. Elaboración: Propia

1. En Millones de pesos. Definición para industria y minería, también existen otras para el sector agropecuario, comercio y servicios.
2. En unidades de fomento (UF). Una UF es aproximadamente US\$24. También existe otra definición por empleo.
3. Existen dos definiciones empleo y activos.
4. Definición únicamente manufacturera. No hay definiciones para comercio o servicios.
5. Existen varios criterios: empleo, ventas y activos. Sin embargo, la definición oficial es por empleo.
6. Existen otras definiciones por inversión y ventas anuales.

b. Número de establecimientos

CUADRO N° 3

Número de empresas

	Micro	Pequeña	Mediana	Grande
1992				
Nº de Empresas	8,648	2,393	1,674	135
Porcentaje	67%	19%	13%	1%
1994				
Nº de Empresas	9,543	2,676	1,807	125
Porcentaje	67%	19%	13%	1%
1995				
Nº de Empresas	10,007	2,801	1,714	124
Porcentaje	68%	19%	12%	1%
1996				
Nº de Empresas	10,7002	3,050	1,786	127
Porcentaje	68%	20%	11%	1%

1997				
Nº de Empresas	11,228	2,081	1,733	137
Porcentaje	69%	19%	11%	1%
1998				
Nº de Empresas	12,005	3,129	1,712	144
Porcentaje	71%	18%	10%	1%
1999				
Nº de Empresas	12,550	3,174	1,778	139
Porcentaje	71%	18%	10%	1%

Fuente: Elaboración propia en base a la Encuesta Económica Anual del MITINCI

c. Participación en el Empleo

CUADRO Nº 4

Participación en el Empleo

	Micro	Pequeña	Mediana	Grande
1992				
Total Per.Ocu.Perm	56,306	34,053	85,987	61,153
Porcentaje	24%	14%	36%	26%
1994				
Total Per.Ocu.Perm	61,891	38,217	91,224	55,285
Porcentaje	25%	16%	37%	22%
1995				
Total Per.Ocu.Perm	65,003	39,669	87,104	52,564
Porcentaje	27%	16%	36%	22%
1996				
Total Per.Ocu.Perm	69,157	43,242	88,584	53,466
Porcentaje	28%	16%	32%	23%

1997				
Total Per.Ocu.Perm	72,467	43,857	84,4474	59,994
Porcentaje	28%	16%	32%	23%
1998				
Total Per.Ocu.Perm	76,998	44,400	85,021	61,170
Porcentaje	29%	16%	32%	23%
1999				
Total Per.Ocu.Perm	78,988	4,894	88,343	63,015
Porcentaje	29%	16%	32%	23%

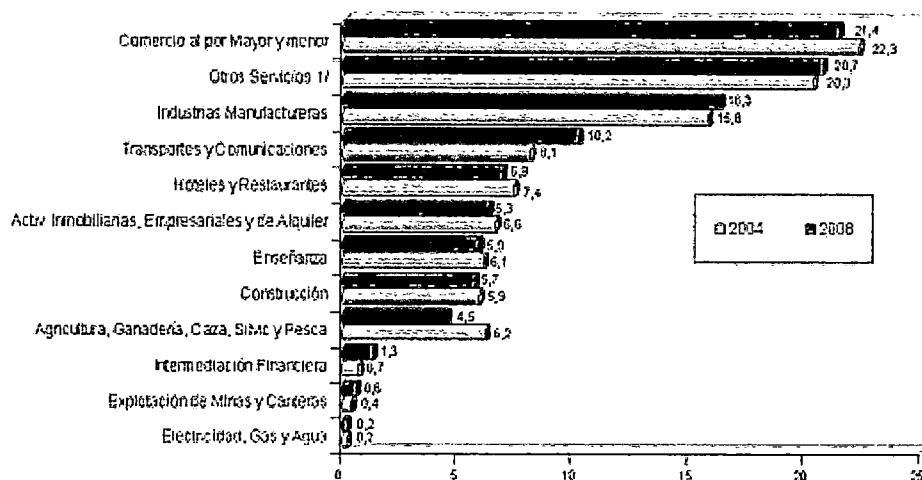
Fuente: Elaboración propia en base a la Encuesta Económica Anual del MITINCI

Anexo 3 Cuadros Estadísticos

Población de Lima Metropolitana

Distrito	Población
San Juan de Lurigancho	765,919
Comas	499,693
San Martín de Porres	469,504
San Juan de Miraflores	375,785
Ate	358,201
Lima Cercado	348,461
Villa el Salvador	337,399
Villa María del Triunfo	335,660
Los Olivos	305,838
Chorrillos	274,309
Santiago de Surco	257,438
La Victoria	232,839
Rimac	214,364
Independencia	208,846
Puente Piedra	173,052
El Agustino	167,715
Sta. Anita	158,133
Carabayllo	151,205
La Molina	140,066
San Miguel	134,103
San Borja	131,060
Lurigancho	127,551
Surquillo	100,784
Breña	96,516
Miraflores	95,473
Pueblo Libre	81,899
Lince	72,604
San Isidro	70,724
Jesús María	68,812
San Luis	60,871
Magdalena	54,539
Barranco	46,915
Chaclacayo	41,780
Ancon	21,746
Sta. Rosa	15,889

LIMA: PEA OCUPADA POR RAMA DE ACTIVIDAD, 2004 Y 2008
(Porcentaje)



1/ Administración pública, defensa, planes de seguridad social, actividades de servicios sociales y de salud, servicios comunitarios, sociales y personales, hogares privados con servicio doméstico y organizaciones y órganos extralocales.

Fuente: INEI-Encuesta Nacional de Hogares (ENAHOG) Continua, 2004 y 2008.

En el 2008, según rama de actividad económica, la PEA ocupada se concentra principalmente en el sector Comercio al por mayor y menor (21,4%), seguido por la otros Servicios (20,7%), Industrias Manufactureras (16,3%) y Transportes y Comunicaciones (10,2%), entre otras actividades. 19

LIMA METROPOLITANA: ESTRUCTURA PORCENTUAL DEL GASTO DE CONSUMO EN LAS ENCUESTAS, SEGÚN GRANDES GRUPOS DE GASTO

GRANDES GRUPOS DE GASTO	ENAPROM 1994 1/	ENAPREF 2009	Diferencia
Total	100,00	100,00	
Alimentos y bebidas	47,65	37,82	-9,7
Vestido y calzado	7,49	5,38	-2,1
Alquiler de vivienda, combustible y electricidad	8,85	9,29	0,4
Muebles, enseres y mantenimiento de la vivienda	4,95	5,75	0,8
Cuidados de la salud y servicios médicos	2,90	3,69	0,8
Transporte y comunicaciones	12,41	16,45	4,0
Esparcimiento, cultura, diversión y serv. de enseñanza	8,82	14,93	6,1
Otros bienes y servicios	7,04	6,69	-0,4

Nota: ENAPROM = Encuesta Nacional de Hogares de Propósitos Múltiples.

ENAPREF = Encuesta Nacional de Presupuestos Familiares

1/ Se revisó la estructura de consumo de hogares 1993-94 mediante el reprocesamiento de la base de datos de la encuesta ENAPROM 1994 como consecuencia de las modificaciones experimentadas en los patrones de consumo, en la oferta de bienes y servicios, así como en las características del sistema de comercialización.

Fuente: Instituto Nacional de Estadística e Informática (INEI) - Dirección Técnica de Indicadores Económicos.

Anexo 4 Funcionamiento de las herramientas

4.1 Funcionamiento de la Herramienta NNClust

Hoja ReadMe

Presenta una descripción en inglés de las instrucciones para el uso de la herramienta.

Paso 1: Ingresar los datos

- (A) Se deben ingresar los datos en la hoja "Data", a partir de la celda 13.
- (B) Las observaciones deben ubicarse en filas y las variables en columnas.
- (C) Por cada columna se debe elegir el tipo apropiado: ("Use" u "Omit").
 - a. Si se quiere que el proceso de clasificación excluya a alguna columna se debe seleccionar la opción = "OMIT".
 - b. Si se quiere incluir la columna en la clasificación, se debe elegir = "USE".

Se pueden ingresar un máximo de 50 variables de agrupación. El aplicativo automáticamente tratará a todas las variables como continuas.

- Debe asegurarse que el número de variables ingresadas en la hoja "Input" sea la misma cantidad de columnas ingresadas en la hoja "Data" de tipo = "USE".
 - Debe asegurarse que el número de observaciones ingresados en la hoja "Input" sea igual o menor a las filas ingresadas en la hoja "Data".
- (D) No puede haber filas o columnas en blanco.
 - (E) Todas las variables a usar en el Clustering deben estar en formato numérico. Las que no cumplan con este requisito serán consideradas valores perdidos. La aplicación podría reemplazarlas por algún dato de la misma columna.

Paso 2: Ingreso de parámetros en la hoja “Input”

(A) Nótese que SOM es una grilla cuadrada conformada por n -neuronas, organizada en n filas y n columnas. Es necesario especificar el valor de n , n debe ser mayor o igual a 2 y menor o igual a 10.

(B) Un ciclo consiste en una presentación de todas las observaciones en el mapa. Por tal motivo es necesario especificar la cantidad de ciclos, es decir la cantidad de veces que se presentarán las observaciones al mapa neuronal. La aplicación acepta como mínimo 2 ciclos.

(C) En cada ciclo todas las observaciones serán presentadas. El orden de presentación puede ser al azar o en el orden en el que fueron ingresados los datos en la hoja “Data”. Se debe seleccionar si se quiere respetar este orden o dejarlo al azar.

(D) Se debe tener presente que el valor final del parámetro de aprendizaje debe ser menor que el valor inicial y ambos valores deben ser mayor o igual a 0 y menor o igual a 1.

(E) Considerar que el valor final de Sigma es menor que el valor inicial y que ambos valores deben ser mayor o igual al 0% y menor o igual al 100%.

(F) A medida que el entrenamiento de la red progresa, ambos parámetros de aprendizaje y de Sigma decrecen desde el valor inicial hacia el valor final, por lo que se debe seleccionar el rango decreciente, ya sea en forma lineal o exponencial

Paso 3: Efecto del botón “Build Clusters”

(A) Mientras el mapa se entrena, los datos de las variables se actualizan de manera que cada valor de las variables se transforman en valores entre -1 y 1 . Esto es lo que se llama normalización de los datos. Este proceso puede ser muy largo, en especial en bases de datos con muchas observaciones y variables.

Si se entrena la red con los mismos datos en dos veces sucesivas, se puede cancelar la normalización en la segunda vuelta. La aplicación preguntará si se quiere cancelar esta normalización o no. Cancelar esta tarea ahorra mucho tiempo. La aplicación siempre se ocupará de chequear el número de filas y columnas en los datos para determinar si ésta ha cambiado desde la última vez que se la ejecutó. No chequea los datos individuales de la hoja "Data". De manera que si se está seguro de que los datos han cambiado desde la última vez que se corrió el algoritmo, se debe nuevamente normalizar los datos.

(B) Si se está entrenando la red con las mismas variables y con las mismas dimensiones de mapa, respecto de la última vez que se corrió, la aplicación preguntará si se quiere comenzar con los pesos obtenidos en el anterior procesamiento. El comenzar con los pesos obtenidos anteriormente aporta incrementos en el aprendizaje. Ésta opción permite resguardar los aprendizajes que se fueron acumulando. Si en cambio los datos se han cambiado, se debe volver a configurar las variables junto con su orden de procesamiento para poder reasignar valores a los pesos.

Paso 4: Resultados del Clustering

(A) Los resultados pueden observarse en la hoja "Output". Los datos son de sólo lectura, ya que la planilla está protegida para evitar modificaciones.

(B) La aplicación ofrece la posibilidad de guardar los resultados en una planilla aparte para que el usuario tenga la posibilidad de poder editar sus resultados.

(C) En esta planilla se pueden guardar los datos procesados por el algoritmo, el cluster asignado a cada observación, y los pesos. Además un gráfico será creado para permitir una comparación visual de los resultados de las variables que atraviesan los diferentes clusters.

(D) En la hoja "Weights", un gráfico dará una representación visual de las observaciones que hay en cada porción del mapa.

Hoja Input

Neural Network based Clustering		(Using Kohonen's Self Organizing Maps (SOM))	
Number of observations	1000	Learning parameter (<i>should be >0 and <1</i>)	Start value 0,8
(Needs to be between 5 and 5,000)			End value 0,1
			Decay Exponential
Number of Variables	3	Sigma for the Gaussian neighborhood	
(Needs to be between 3 and 50)		as % of map width (<i>should be > 0% and < 100%</i>)	
Enter <i>n</i> , where <i>n</i> -Square = # neurons in the map	3	Start value 50,0%	
(<i>n</i> needs to be between 2 and 10)		End value 1,0%	
		Decay Exponential	
[Note: By entering <i>n</i> you are specifying that the maximum number of clusters will be at most <i>n</i> -square. e.g. if you enter <i>n</i> = 4, you will get less than or equal to 16 clusters]			
Number of training cycles	50		
(Needs to be between 1 and 1000)			
			Build Clusters
Randomize the order in which inputs are presented to the map ?	No		

En esta hoja se ingresan los parámetros de configuración del funcionamiento del algoritmo. Estos datos deben ser consistentes con los ingresados en la hoja "Data".

Se debe ingresar:

- Número de observaciones. (Valor entre 5 y 5.000)
- Número de variables. (Valor entre 3 y 50)
- Las dimensiones del mapa, valor que será elevado a la potencia 2 (dos) para obtener el total de neuronas del mapa. (Entre 2 y 100)
- Número de ciclos de entrenamiento. (Entre 1 y 1.000)
- Parámetros de aprendizaje (mayor a 0 y menor que 1)

- Valor inicial
- Valor final
- Forma del decrecimiento = Exponencial o Lineal
- Valor de Sigma para la vecindad Gaussiana, como porcentaje del ancho del mapa.
 - Valor inicial
 - Valor final
 - Forma de decrecimiento = Exponencial o Lineal

Hoja Data

	A	B	C	D	E	F	G	H	I
1	Enter your Data in this sheet								
2	<i>Instructions:</i>								
3	Start Entering your data from cell C13.			Specify variable names in row 11.					
4	Specify variable type in row 10.			Use - to use variable for clustering. Omit - not to use the variable for clustering					
5	Please make sure that there are <u>no blank row or column</u> in your data.								
6	The variables that you use for clustering <u>needs to be numeric</u>								
7	Any <u>non-number</u> in your data will be treated as <u>missing value</u> and will be <u>replaced</u> by the respective <u>column mean</u>								
8									
9									
10		Use	Use	Omit	Omit	Omit	Omit	Omit	Omit
11		Cantidad	Precio	Total_Factura					
13		7	7,000	49,000					
14		7	7,000	49,000					
15		7	7,000	49,000					
16		7	7,000	49,000					
17		7	7,000	49,000					
18		7	7,000	49,000					
19		7	7,000	49,000					
20		7	7,000	49,000					
21		7	7,000	49,000					
22		7	7,000	49,000					
23		7	7,000	49,000					
24		7	7,000	49,000					
25		7	7,000	49,000					
26		7	7,000	49,000					
27		7	7,000	49,000					
28		7	7,000	49,000					
29		7	7,000	49,000					
30		7	7,000	49,000					
31		7	7,000	49,000					

Los datos deben ingresarse a partir de la celda C13

Los nombres de las variables deben ingresarse en la fila 11.

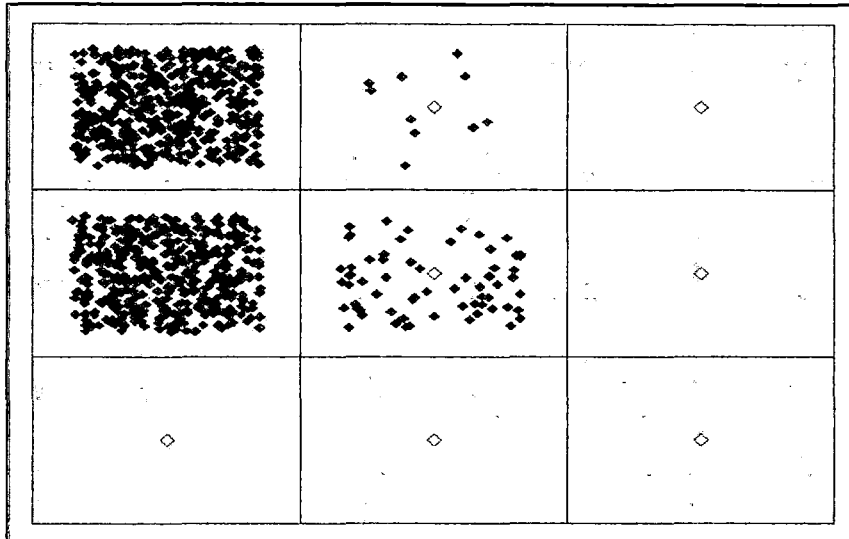
El tipo de variable debe especificarse en la fila 10 Para que la variable sea incluida por el algoritmo se debe indicar la opción "USE", para que sea ignorada "OMIT".

No debe haber filas o columnas con datos en blanco. Las variables que se

usan para clasificar deben estar en formato numérico.

Los datos que no están en formato numérico serán considerados como datos perdidos y serán reemplazados por un valor de la misma columna.

Hoja Weihgts



Visualiza el mapa bidimensional. Las observaciones se van ubicando en cada cluster a medida que la aplicación se está ejecutando.

Hoja Output

En esta hoja presenta los resultados del algoritmo.

Informa:

- El número de variables usadas para clasificar
- El número de observaciones usadas para clasificar
- Cantidad de grupos (clusters)
- Tabla "Cluster Assignment" (Asignación de clusters): muestra por cada observación, (representada por una ID) el número de cluster asignado
- Tabla "Clusters Size" (Tamaño de los clusters): muestra la cantidad de

observaciones encontradas en cada uno de los clusters o grupos.

- Tabla "Cluster Position on the grid" (Posición de cada cluster dentro de la grilla). Tabla de doble entrada donde se indica el número de fila y de columna que le corresponde a cada cluster.
- Tabla "Cluster Means" (Promedio de los clusters): Tabla de doble entrada donde se indica los valores promedio para la totalidad de los datos, y para cada uno de los clusters.
- Tabla "Cluster Variantes" (Varianza de los clusters): Tabla de doble entrada donde se indica la varianza para la totalidad de los datos y para cada uno de los clusters.

Hay una nota, que indica que aquellas celdas que estén de color azul en la tabla "Clustes Means" y "Clusters VariANCES", advierte la pérdida de valores para esa variable dentro del cluster. Todos los valores perdidos fueron reemplazados por valores medios o varianzas del cluster.

	E	F	G	H	I	J	K	L	M	N	O	P	Q				
Clustering using Self Organizing Maps						Note: Cells marked with blue (if any) in the Cluster Means and Variances tables denote that there are some missing values for that variable in that cluster. All missing values have been replaced by overall means for computation of cluster means and variances											
Number of variables used for clustering						3											
Number of observations used for clustering						1000											
Number of Clusters						3											
Cluster Assignment						Cluster Sizes											
Observation ID	Cluster ID					Cluster 1			Cluster 2			Cluster 3					
1	1					28			61			911					
2	3					Cluster Position on the grid											
3	3					Cluster 1			Cluster 2			Cluster 3					
4	3					Row	1	1	1								
5	3					Column	1	3	4								
6	3					Cluster Means											
7	3					Overall			Cluster 1			Cluster 2			Cluster 3		
8	3					Cantidad	5,1	7,0	7,9	4,9							
9	3					Precio	4,0	7,0	5,5	3,8							
10	3					Total_Factura	20,0	49,0	43,1	17,6							
11	3					Cluster Variances											
12	3					Overall			Cluster 1			Cluster 2			Cluster 3		
13	3					Cantidad	8,5	0,0	1,0	8,6							
14	3					Precio	1,4	0,0	0,3	1,1							
15	3					Total_Factura	144,7	0,0	4,5	91,1							
16	3																
17	3																
18	3																
19	3																

Hoja Junk y Hoja Plot

En esta hoja la aplicación guarda datos de soporte para generar los resultados

4.2 Funcionamiento de la herramienta NNClass

Se trata de una implementación en Excel, en la cual se pueden ver diferentes hojas para completar los datos que servirán para definir y construir la red (Hoja UserInput), ingresar los datos con los que el algoritmo definirá los pesos (Hoja Data). Además, presenta la hoja Calc, en la cual, se muestra el conjunto de pesos finales una vez construido el modelo. También mostrará los valores de de los errores cuadráticos del entrenamiento y la validación en la hoja denominada Output.

Hoja ReadMe

Esta hoja presenta una descripción en inglés de las instrucciones para el uso de la herramienta.

Paso 1: Ingresar los datos

(A) Ingresar los datos en la hoja Data, empezando por la celda AC 105

(B) Las observaciones deben ubicarse en filas y las variables en columnas

(C) Elegir sobre cada columna, el tipo apropiado (Omit, Output, Cont, Cat)

- Si no desea que la columna sea sometida al modelo, seleccione = OMIT
- Si desea que la columna sea tratada como categoría, seleccione =

CAT

- Si desea que la columna sea tratada como continua, seleccione = CONT
- Si desea que la columna represente la salida, seleccione = OUTPUT

Para clasificar problemas, se requiere una variable de salida. La aplicación, tratará automáticamente a esta columna, como una variable de salida (OUTPUT).

Se puede ingresar como máximo 50 variables de entrada (INPUT) de las cuales 40 pueden ser variables de tipo categoría (CAT). Debe asegurarse que la cantidad de columnas de entrada y las de tipo categoría coincidan con los datos registrados en la hoja UserInput

(D) La hoja de datos no puede contener celdas con datos en blanco

(E) Entradas tipo del tipo Continuas:

- Todo dato no numérico ingresado en este tipo serán considerados como Datos Perdidos
- La aplicación los reemplazará por algún valor promedio de su misma columna

(F) Entrada tipo Categoría

- Toda celda vacía o conteniendo Error de Excel, será considerado como Dato Perdido.
- La aplicación reemplazará a este valor, por el dato de mayor frecuencia dentro de su categoría.
- Los nombres de las categorías no detectarán diferencias entre nombre como: GOOD, Good, GoOd, good. Todos serán considerados como misma categoría.
- Puede haber como mínimo 2 observaciones en cada categoría. Si

hay una sola, se la debe eliminar o renombrarla y ubicarla en otra categoría.

Paso 2: Completar datos en la planilla Input

- (A) Completar los datos
- (B) Considerar que los valores ingresados están dentro de los rangos que
- (C) Presionar en el botón "Build Model" para ejecutar la aplicación

Paso 3: Resultados obtenidos

(A) Un conjunto de Red Neuronal es básicamente un conjunto de pesos distribuidos entre las capas de la red. Al finalizar la aplicación del modelo obtendrá el conjunto de pesos finales que serán guardados en la hoja "Calc".

(B) La hoja "Output" de este archivo, mostrará los valores del entrenamiento y la validación, mientras el entrenamiento está en progreso de ejecución. Los gráficos del entrenamiento y la validación se muestran en la hoja "Output". Estos pueden ser modificados o borrados.

(C) Si en la hoja "UserInput" se aceptó guardar los resultados en otra hoja, la aplicación creará una nueva hoja conteniendo las entradas del modelo, los datos ingresados, y el modelo ajustado. Este archivo podrá ser utilizado para realizar clasificaciones con cualquier dato que se ingrese.

Paso 4: Cuadro de perfiles

Una vez que está construido el modelo se pueden estudiar los perfiles en la hoja "Profiles". Si la variable de clase tenía k categorías, habrán k curvas por cada perfil. Por cada nueva observación, el modelo estará en condiciones de

predecir k puntos por cada k categorías de la variable de clase. El modelo podrá predecir la categoría de clase como la de mayor puntaje. Cada uno de estos puntajes es obtenido en función de su pronóstico. Si se tienen p predicciones, entonces el puntaje para cada categoría es la superficie dimensional de p . Si p es 2 o menor a 2, no es posible mostrar la superficie gráficamente. El cuadro de perfiles es la mejor forma de visualizar la superficie ajustada. Si se va modificando un solo pronóstico entre dos valores y se van dejando todos los demás valores fijos con algún valor preestablecido, obtendremos el cuadro de perfiles, que es realmente una sección representativa de una sola dimensión de la más alta superficie. En la hoja "Profiles" se puede especificar cual pronóstico modificar y que valores de los demás predictores quedarán fijos. Presionando sobre el botón "Create Profile" se podrá generar el perfil. Si el predictor elegido es de tipo "Categorical", entonces la demás información (puntos a generar, valores de inicio y final), serán ignorados y el gráfico mostrará los puntajes de cada categoría del predictor que fueron elegidos para variar.

El cuadro de perfiles permite estudiar las siguientes cuestiones:

1. Dado un predictor X , qué rango de valores estará asociados a la clase 1, clase 2, clase k , etc. (Por ejemplo, se puede encontrar que el puntaje para la clase 1 es mayor cuando X es pequeño; cuando X está en su rango medio el puntaje de la clase 2 es el más alto, y para muchos valores altos de X el puntaje de la clase 3 es el más alto, etc.
2. El gráfico de perfiles permite además estudiar la interacción entre predictores. Supongamos que estamos estudiando los perfiles de la clase 1 y dos predictores son X y Z . Supongamos que estamos observando el perfil dejando a Z con su valor fijo en 1 y variando a X entre -10 y 10. Luego dejamos fijo a Z en 2 en lugar de 1 y variamos a X entre -10 y 10. Si la forma de los perfiles es totalmente diferente

entonces quiere decir que los predictores X y Z interaccionan. Es decir, que el efecto de X sobre la variable de clase no es la misma para todos los valores de Z . Para estudiar los efectos de X , importa como Z esté configurado.

Paso 5: Pendiente del Gráfico

Una vez que el modelo está construido, se puede estudiar la pendiente del gráfico para una clase de categoría en particular, en la hoja "LiftChart". Este muestra como la pendiente del gráfico está construida. Suponga que estamos observando la pendiente del gráfico de la categoría k . Observamos todos los puntajes (generados por el modelo) para la categoría k . Ordenamos los datos en forma descendente según estos puntajes. Si el modelo es bueno, podemos esperar que las observaciones que recibieron los puntajes más altos, puedan en realidad pertenecer a la categoría k . Por lo que alrededor de las pocas observaciones más altas, el porcentaje de la categoría actual igual a k , será el más alto. Para N más altas observaciones, se computa el % con la categoría k y el % con la categoría *no* k . Se desarrolla el gráfico con estos dos valores, $X = \% \text{ not-}k$ versus $Y = \% k$ para generar la pendiente de la curva.

La pendiente del gráfico aporta la siguiente información:

1. Qué es el poder discriminatorio del modelo. Si al menos queremos capturar, por ejemplo el 90% de las observaciones de la categoría k exactamente usando el modelo, entonces que porcentaje de no- k observaciones el modelo captura por error. La pendiente de la curva siempre comienza en 0%, 0% y termina en 100%, 100%. La diagonal que nace en 0%,0% y termina en 100%,100% debe estar siempre dibujada como referencia. Cuando mejor es el modelo, más alto resultará la pendiente de la

curva sobre la línea diagonal. El mejor escenario posible se da cuando el 100 % pertenece a la categoría k sin captura un $no-k$. En este caso la curva comienza en 0%, 0% crece verticalmente hacia 0%, 100% y luego se dirige horizontalmente hacia 100%, 100%.

Otros ítems a considerar:

Pesos iniciales:

Para entrenar el modelo necesitamos comenzar con un conjunto de valores iniciales para los pesos de la red. Por defecto los pesos iniciales deben ser valores aleatorios entre $-w$ y w , donde w es un número entre 0 y 1, que debe especificarse en la hoja UserInput.

(A) Una vez construido el modelo, los pesos finales son guardados en la hoja "Calc". La próxima vez que se quiera entrenar el modelo con la misma arquitectura y los mismos datos, la aplicación preguntará si se quiere comenzar con los pesos guardados en la hoja "Calc". Si se responde que Si, entonces estos pesos serán utilizados; pero si se responde No se reinicializarán con valores aleatorios.

(B) En lugar de empezar con pesos aleatorios, se puede empezar con los propios pesos elegidos. Especificar la elección de especificar los pesos, es una tarea no trivial para esta aplicación. Aquí se explica como hacer.

- Especificar las entradas en la hoja "Input" y especificar el número de ciclos de entrenamiento como 0.
- Esto configurará la hoja "Calc" para que no realice el entrenamiento.

- En la hoja "Calc" se debe escribir la elección de pesos en el lugar apropiado de la matriz de pesos.

Se debe volver a la hoja "UserInput" y especificar el número de ciclos deseado y presionar el botón "Build Model". Cuando la aplicación pregunte si se quiere usar los pesos guardados, se debe responder Si. Ahora la red se entrenará con los pesos especificados por el usuario.

Hoja UserInput

	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	
102	Network Architecture Options															
103																
104	Number of inputs (between 2 and 50)					4										
105	Number of Hidden Layers (1 or 2)					2					Hidden Layer sizes (Maximum 20)					
106											Hidden 1		Hidden 2			
107											15		5			
108	Learning parameter (between 0 and 1)					0.7					Initial Wt Range (0 +/- w): w =					
109											0.7					
110	Momentum (between 0 and 1)					0.1										
111																
112																
113	Training Options															
114	Total #rows in your data (Minimum 10)					1000					No. of Training cycles (Maximum 500)					
115											100					
116	Present inputs in Random order while Training?					NO					Training Mode (Batch or Sequential)					
117											Sequential					
118																
119	Saving Network Weights					With least Training Error										
120																
121	Training / Validation Set					Use whole data as training set										
122																
123	If you want to partition, how do you want to select the Validation set?															
124	Please choose one option					1					Option 1: Randomly select					
125	Please fill up the input necessary for the selected option										10%					
126											of data as Validation set (between 1% and 50%)					
127											30					
128											rows of the data as validation set					
129	Save model in a separate workbook?					NO										

En esta hoja se ingresan los parámetros necesarios para configurar el funcionamiento de la aplicación. Estos datos deben ser consistentes con los

ingresados en la Hoja "Data".

Deben ingresarse:

Opciones de arquitectura de red

- Número de observaciones (entre 2 y 50).
- Número de capas ocultas (entre 1 y 2). – Tamaño de las capas ocultas (Máx. 20)
- Parámetro de aprendizaje (0 ó 1) – Rango inicial de peso Wt (0 +/-w): $w =$ (valor ingresado)

Si por ejemplo se dijo que para este patrón en particular se necesitó un ajuste del 10%, se tendría que realizar entonces la siguiente operación:

$$\text{NuevoPeso} = \text{PesoAnterior} * \text{Ajuste} * \text{ValordeAprendizaje}$$

El valor de aprendizaje es de importancia en un único patrón o patrón aislado, por ejemplo si se estableciera en 0.01, tomaría 100 patrones el llegar a un ajuste del 10%. .

- Momento (0 ó 1)

Es algo que se necesita y puede agilizar los cálculos de manera significativa. Supongamos que tenemos 100 patrones y que nos damos cuenta que cada uno se mueve

el 0.01% hacia algún valor, sería mucho mejor que se moviera de manera más rápida (siempre hacia la misma dirección).

Opciones de entrenamiento

- Cantidad total de filas de datos (min. 10) – Número de ciclos de entrenamiento (máx. 50)

- Seleccionar si se presentan entradas en orden aleatorio durante el entrenamiento (Si o No) – Modo de entrenamiento (Serial o Secuencial)

Modo de guardar los pesos de la red Se debe seleccionar entre las siguientes tres opciones:

- Al final del último ciclo,
- Con el último error validado
- Con el último error de entrenamiento

Configuración del entrenamiento y validación. Se debe seleccionar entre las dos alternativas:

- Usar una parte de la base de datos
- Usar toda la base de datos

Si se desea usar sólo una parte, se debe seleccionar el modo de validación. Se puede optar por:

- 1 (uno) significa que se seleccionará un porcentaje variable de datos a validar (entre 1% y 50%).
- 2 (dos) significa que se usarán las últimas filas de datos. (Cantidad de filas a ingresar)

Indicar si se quiere guardar los resultados del modelo en una hoja aparte (Si o No)

Hoja Data

Los datos debe ingresarse a partir de la celda AC105 Los nombres de las

variables deben ingresarse en la fila 103. Debe verificarse que la fila 104 esté vacía.

El tipo de variable debe especificarse en la fila 102. CONT (para indicar que la columna es de tipo Continua), CAT (para indicar que una columna es de tipo Categoría), OUTPUT (para dejar la columna fuera de los límites de la aplicación) y OMIT (si no se quiere usar el nombre específico de variable en la fila 103)

85	Enter your Data in this sheet						
86	Instructions: Start Entering your data from cell AC105 .			Specify variable name in row 103.			
87	Make sure that the row 104 is blank.						
88	Specify variable type in row 102.						
89	Cont - for continuous Input,						
90	Cat - for Categorical Input,						
91	Output -for Output var.						
92	Omit - if you don't want to use the variable in the n Specify variable names in row 103						
93	For each continuous Input, there will be 1 neuron in Input Layer.						
94	For Each categorical Input with K levels, there will be K neurons in Input Layer						
95	Please make sure that there are <u>no more than 50 neurons</u> in Input Layer.						
96	There should be <u>exactly 1 Output variable</u> - application will treat it as Categorical.						
97	There should be <u>no more than 40</u> Categorical Variables.						
98							
99							
100							
101							
102	Var Type	Omit	Output	Cont	Cont	Cont	Cont
103	Var Name	IDENTIFICA	PRODUCTO	FRANJA	CANTIDAD	CLIENTE	MOZO
105		6 pocillo		2	2	1	2 A
106		6 pocillo		2	2	1	1 B
107		7 Café Morgana		2	2	1	1 C
108		7 Café Morgana		2	2	1	2 A
109		7 Café Morgana		2	2	1	2 B
110		6 pocillo		2	2	1	1 C
111		7 Café Morgana		2	2	1	1 A
112		6 pocillo		2	2	1	2 B

Por cada entrada de tipo continua, habrá una neurona en la línea de entrada
 Por cada entrada de tipo categórica con k niveles, debe haber k neuronas en la línea de entrada.

No debe haber más de 50 neuronas en la línea de entrada. Debe haber una sola variable de salida, la cual la aplicación la tratará como atributo de tipo

categorico. No puede haber más 40 variables de tipo categorico.

Hoja Calc

Neural Network Model for Classification							Created On:	27-dic-10
% MissClass.(Training)		69.00%		% MissClass.(Validation)				
Number of Hidden Layers		2						
Layer Sizes		4		15		5 15		
True Output (if available)								
Model Output		merienda						
Raw Input		Bias	Cont	Cont	Cont	Cont		
		1	FRANJA	CANTIDAD	CLIENTE	MOZO		
		1	3.7210	10.9000	1.7380	1.4670		
Transformed Input		Bias	FRANJA	D	CLIENTE	MOZO		
		1	0.5442	0.8999	0.2461	0.4670		
Hdn1_bias		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
Hdn1_Nrn1		4.9584	-28.1124	6.6433	-0.8665	-15.9154	-12.0074	
Hdn1_Nrn2		5.1166	-24.5650	20.4610	-3.5503	-1.1386	8.7565	
Hdn1_Nrn3		-1.6201	-16.1709	7.2333	-0.4135	-1.2388	-4.5910	
Hdn1_Nrn4		9.7442	5.7026	-16.4125	10.0806	0.9199	0.9875	
Hdn1_Nrn5		-20.7877	9.1606	10.3818	-8.4912	-3.5493	-10.2066	
Hdn1_Nrn6		-4.2842	33.2926	-15.0000	10.6938	-2.8598	1.6302	
Hdn1_Nrn7		-20.2013	14.8960	-1.3733	-2.8036	-3.0455	-15.4430	
Hdn1_Nrn8		-13.7757	7.0072	1.2188	-3.7571	-2.2947	-10.8617	
Hdn1_Nrn9		1.0478	-36.2535	27.4181	-9.7778	-8.2680	-0.2726	
Hdn1_Nrn10		-5.2893	10.3367	33.2277	-6.7482	0.0602	28.6059	
Hdn1_Nrn11		-17.1608	26.4948	9.7013	-4.9855	-0.1539	4.6892	
Hdn1_Nrn12		12.5287	1.3484	-17.9647	6.8918	1.6591	-0.4363	
Hdn1_Nrn13		-11.4929	5.8268	-2.6595	1.1313	-2.4507	-11.5814	
Hdn1_Nrn14		-5.9996	-3.1689	1.0085	-0.4758	2.5234	-5.7551	
Hdn1_Nrn15		5.1470	-1.6644	-0.3367	0.1438	1.9501	4.8844	

Enter your inputs in the range AG112:AJ112 - the cells marked in green.

Esta hoja es la que realiza los cálculos del modelo e informa de ellos.

Visualiza la siguiente información:

Porcentaje de datos que durante el entrenamiento quedaron fuera de la clasificación y porcentaje de datos que quedaron fuera de la clasificación

durante la validación.

Número de capas ocultas seleccionadas previamente

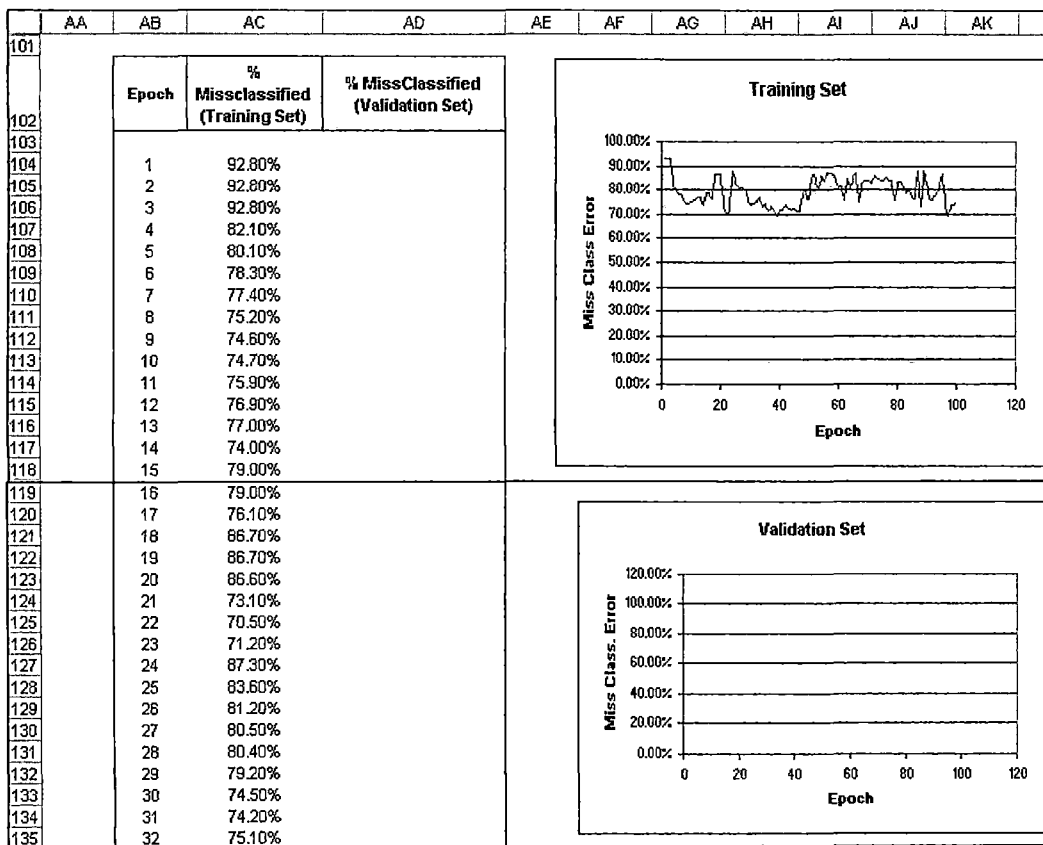
Tamaño de las capas (la de entrada, salida y las ocultas)

Salidas de tipo verdaderas (en caso de esta disponible)

Resultado del modelo. (Este dato se modifica a partir de los cambios en los datos introducidos en las celdas AG112 hasta AJ112)

El resto de los datos dan soporte al modelo para poder dar un resultado.

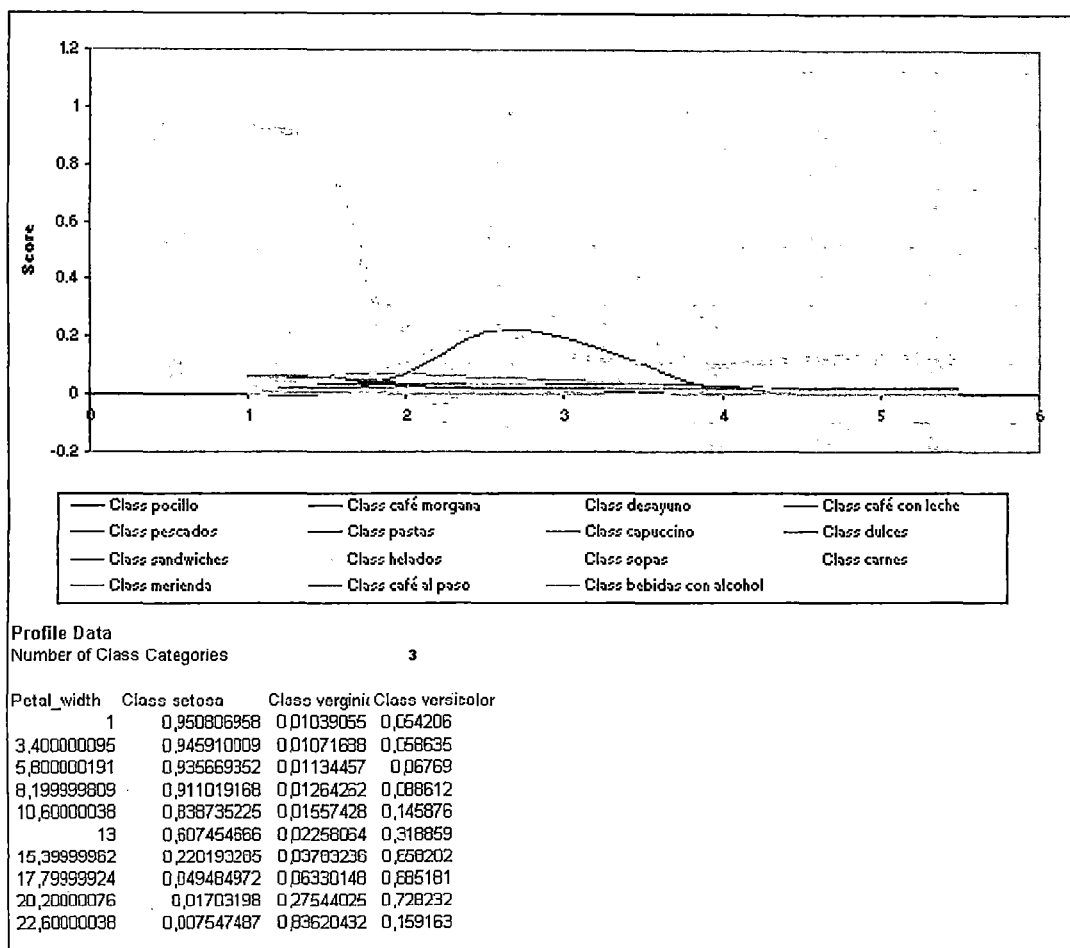
Hoja Output



La tabla muestra los porcentajes no incluidos en la clasificación durante el

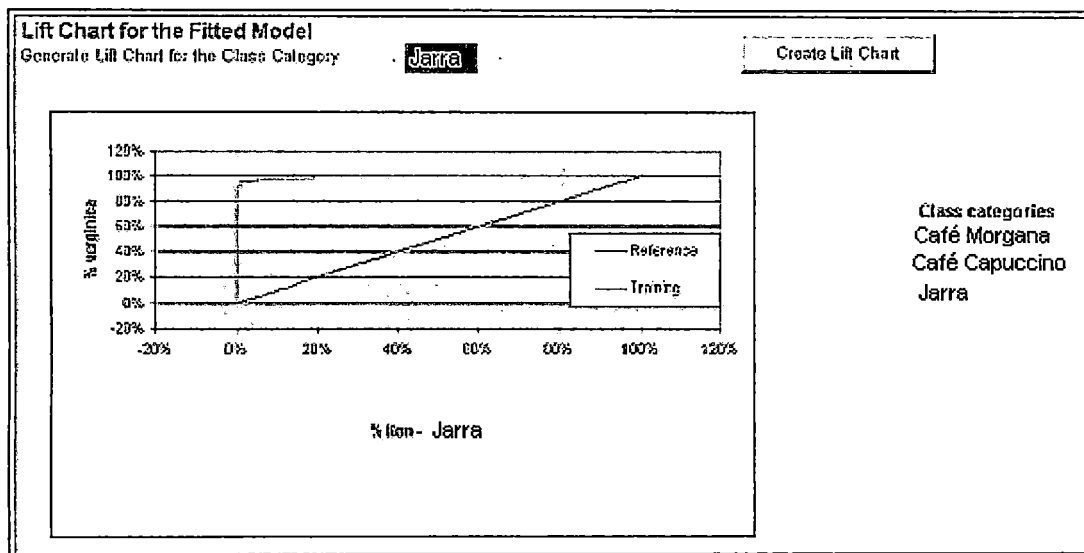
entrenamiento y en la validación, cada una con sus respectivos gráficos. En los resultados obtenidos se observa que a medida que el modelo se entrena, disminuye la cantidad de datos que quedan fuera de la clasificación.

Hoja Profile



Presionando el botón "Create Profile", se genera el perfil ajustado del modelo. Para ello se deben ingresar algunos valores, que este proceso tiene en cuenta:

Hoja LifChart



Anexo 5 Otras Herramientas Disponibles

La siguiente es una lista con el nombre de varios Software comerciales de redes neuronales.

Software	Compañía	Site
NeurOn-Line	Gensym Corporation 125 Cambridge Park Drive. Cambridge, Massachussetts 02140 <i>Representante en</i> <i>Argentina.</i> Soteica S.R.L.: Av. Alvarez Thomas 796 - 3° C, (1427) Buenos Aires, Argentina	http://www.gensym.com http://www.soteica.com.ar
HNeT (Holographic/Quantum Neuronal Technology).	AND Corporation (212) 279-3833 Nueva York, (416) 920-8260 Toronto	http://www.andcorporation.com
NeuroMonitor Pro.	ERA Technology Cleeve Road, Leatherhead, Surrey, KT22 7SA UK.	http://www.era.co.uk
NeuroShell Predictor.	Ward Systems Group, Inc. Executive Park West, 5 Hillcrest Dr. Federick , MD 21703	http://www.warsystems.com http://www.neuroshell.com
NeuroDynamX®	DynaMind®	http://neurodynamx.com
NeuroSolutions	NeuroDimension Inc.	http://www.nd.com
Neural OptiMatch™	Neuralr	http://www.neural.com