



**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**Instituto de Ciencias Matemáticas**

**“Aplicación de Redes Neuronales para el Análisis  
de la Cartera Vencida de Empresas, con el fin  
de Mejorar Su Rentabilidad”**

**TESIS DE GRADO**

**Previa a la obtención del Título de:**

**INGENIERO EN ESTADISTICA INFORMATICA**

**Presentada por:**

**Luis Alberto Muñoz Santa Cruz**



**GUAYAQUIL - ECUADOR**

**2002**

# DEDICATORIA

A mis padres, hermanos y a mi novia,  
quienes me apoyaron con abnegación y  
esfuerzo para el feliz término del  
presente trabajo y de mis estudios.

## **DECLARACION EXPRESA**

“La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

---

**Luis Alberto Muñoz Santa Cruz**

## TRIBUNAL DE GRADUACION

---

Mat. Jorge Medina  
DIRECTOR DEL I.C.M.

---

Ing. Carlos Jordán Villamar  
DIRECTOR DE TESIS

---

Ing. Mario Castro Mora  
VOCAL

---

Ing. Edison Del Rosario  
VOCAL

## RESUMEN

La presente aplicación establece una metodología a seguir para la realización del análisis crediticio de un solicitante de un servicio. He tomado como caso particular la compañía operadora de telefonía celular, BellSouth Guayaquil, con su producto "crédito", ya que sus analistas están tomando decisiones crediticias basándose en el factor de ingresos futuros, antes que en la liquidez del cliente.

BellSouth necesita automatizar el proceso de inspección de Cuentas, ya que es un procedimiento muy lento y costoso en términos de recursos utilizados; dado que incluye tres etapas íntimamente relacionadas: a) Obtención de información del solicitante, b) Análisis de esa información para determinar el valor crediticio del solicitante, c) Toma de decisión sobre el crédito. La

decisión sobre el crédito a su vez establece si el crédito debe otorgarse y el plan tarifario que puede acceder.

Se requiere de estructuras de redes neuronales para lograr la automatización de la verificación de calidad crediticia de las Cuentas aceptadas, las que utilizando el concepto de sistema automático para el control de las Cuentas por Cobrar reducirán la cantidad de tiempo que se usa en éste análisis.

## INDICE GENERAL

|   | Pág. |
|---|------|
| INTRODUCCIÓN.....   | 1    |
| I. ANTECEDENTES DE LA CARTERA VENCIDA DE BELLSOUTH,<br>PERIODO JUNIO DEL 2000 – JUNIO DEL 2001..... | 4    |
| 1.1. Proceso de evaluación de los créditos individuales de la<br>operadora BellSouth Guayaquil..... | 6    |
| 1.2. Valoración crediticia de los clientes de BellSouth.....  | 9    |
| 1.3. Factores de evaluación a los clientes de BellSouth.....  | 13   |
| 1.4. Decisión de crédito en los clientes de BellSouth.....  | 16   |
| 1.5. La cartera vencida desde el punto de vista de Redes<br>Neuronales.....                         | 17   |
|   |      |
| II. BREVE DESCRIPCIÓN DEL ANÁLISIS DISCRIMINANTE.....   | 20   |
| 2.1. Introducción.....  | 20   |
| 2.2. Separación y clasificación para dos poblaciones.....   | 22   |
| 2.2.1. Clasificación de dos poblaciones.....  | 26   |
| 2.3. Observaciones.....   | 35   |
| 2.3.1. Incluyendo variables cualitativas.....   | 35   |
| 2.3.2. Observaciones finales.....   | 37   |
|   |      |

|  |    |
|--|----|
| III. REDES NEURONALES RATIFICALES.....                                 | 39 |
| 3.1. Introducción.....   | 39 |
| 3.2. La neurocomputación.....  | 40 |
| 3.2.1. Analogía con el cerebro.....                                    | 40 |
| 3.2.2. Redes Neuronales.....   | 42 |
| 3.2.3. Operación de las redes.....                                     | 44 |
| 3.2.3.1. Operación de las capas.....                                   | 48 |
| 3.2.4. Que hace a la neurocomputación diferente?.....                  | 49 |
| 3.2.4.1. Aprendizaje a través de ejemplos.....                         | 50 |
| 3.2.4.2. Memoria de Asociación Distribuida.....                        | 52 |
| 3.2.4.3. Tolerancia a las fallas.....                                  | 54 |
| 3.2.4.4. Reconocimiento de patrones.....                               | 55 |
| 3.2.4.5. Poder de Síntesis.....  | 57 |
| 3.2.5. El Perceptrón de Frank Rosenblatt.....                          | 58 |
| 3.2.6. Propagación hacia atrás (Back-Propagation).....                 | 63 |
| 3.2.6.1. La red de Propagación hacia atrás.....                        | 64 |
| 3.2.6.2. Propagando hacia atrás el error local.....                    | 68 |
| 3.2.6.3. Minimizando el error local.....                               | 70 |
| 3.2.6.4. La función de error global.....                               | 72 |
| 3.2.6.5. Resumen del algoritmo estándar de propagación<br>inversa..... | 74 |



|   |     |
|---|-----|
| 3.2.6.6. Variaciones al algoritmo estándar.....   | 74  |
| 3.3. Observaciones.....   | 80  |
|   |     |
| IV. ANÁLISIS Y APLICACIÓN DE UN MODELO DE REDES<br>NEURONALES UTILIZANDO UN SOFTWARE PARA<br>RESOLVER EL PROBLEMA DE CARTERA VENCIDA..... | 82  |
| 4.1. Introducción.....  | 82  |
| 4.2. Sobre los datos.....   | 83  |
| 4.3. Acerca del Software utilizado.....   | 85  |
| 4.4. Proceso de Aplicación del modelo sobre los datos.....  | 86  |
| 4.4.1. Creación de la red.....  | 87  |
| 4.4.2. Importancia de los datos.....  | 94  |
| 4.4.3. Entrenamiento de la red.....   | 96  |
| 4.5. Obtención de los resultados.....   | 86  |
| 4.6. Observación.....   | 101 |
| 4.6.1. Sobre el conjunto de datos.....  | 101 |
| 4.6.2. Sobre el uso práctico de la red.....   | 103 |
|   |     |
| CONCLUSIONES Y RECOMENDACIONES  |     |
| ANEXO   |     |
| BIBLIOGRAFÍA  |     |

# INDICE DE FIGURAS

|            |   | Pág. |
|------------|---|------|
| Figura 1.1 | Monto de cartera vencida del Periodo Junio del 2000 a Junio del 2001.....             | 5    |
| Figura 1.2 | Porcentaje de Recuperación de la Cartera Vencida.....                                 | 6    |
| Figura 2.1 | Ingreso y tamaño de lote para poseedores y no poseedores de cortadoras de césped..... | 32   |
| Figura 3.1 | Neurona Biológica.....  | 41   |
| Figura 3.2 | Un elemento de proceso o nodo.....  | 43   |
| Figura 3.3 | Un Perceptrón.....  | 58   |
| Figura 3.4 | Problema XOR.....   | 61   |
| Figura 3.5 | Una solución para el problema XOR utilizando dos perceptrones.....                    | 62   |
| Figura 3.6 | Red típica de Back-Propagation.....   | 65   |
| Figura 3.7 | Elemento de proceso típico en Back-Propagation.....                                   | 67   |
| Figura 3.8 | Funciones de transferencia para propagación hacia atrás                               | 78   |
| Figura 4.1 | Estructura de la red neuronal que resuelve el problema...                             | 87   |
| Figura 4.2 | Parametrización en la capa de entrada.....  | 90   |

|            |   |    |
|------------|---|----|
| Figura 4.3 | Parametrización de la red en la capa oculta.....          | 91 |
| Figura 4.4 | Parametrización de la red en la capa de salida.....       | 92 |
| Figura 4.5 | Muestra de las hojas de datos de ThinksPro.....           | 95 |
| Figura 4.6 | Evolución del error global conforme la red es entrenada.. | 97 |
| Figura 4.7 | Visor de la evolución del error cada 100 iteraciones..... | 98 |
| Figura 4.8 | Resultados de ThinksPro al cabo de 933 iteraciones.....   | 99 |

# INDICE DE TABLAS

|           |   | Pág. |
|-----------|---|------|
| Tabla 2.1 | Ejemplos de Clasificación y Discriminación..... | 24   |
| Tabla 2.2 | Clasificación de Familias.....                  | 31   |

# SIMBOLOGIA

|            |                |
|------------|----------------|
| $\pi$      | Pi             |
| $\alpha$   | Alfa           |
| $\beta$    | Beta           |
| $\Sigma$   | Sumatoria      |
| $f(x)$     | Función de x   |
| $\nabla$   | Diferenciación |
| $\partial$ | Delta          |

## INTRODUCCIÓN

El presente trabajo es la aplicación de un modelo de Redes Neuronales, utilizando un software para resolver el problema de la cartera vencida en una empresa. En él se han analizado cuáles son las características que están íntimamente relacionadas con el riesgo de que se conviertan en cuentas perdidas, lo que se logra sobre la base de una solución factible en la aproximación de Redes Neuronales.

En ésta aplicación se establece una técnica a seguir para la realización del análisis crediticio de un solicitante de un servicio. He tomado como caso particular la compañía operadora de telefonía celular, BellSouth Guayaquil, con su producto "crédito", ya que sus analistas están tomando decisiones crediticias sobre el factor de ingresos futuros, antes que la liquidez del cliente.

BellSouth necesita automatizar el proceso de inspección de Cuentas, ya que es un procedimiento muy lento y costoso en términos de recursos utilizados; dado que incluye tres etapas íntimamente relacionadas: a) Obtención de información del solicitante, b) Análisis de esa información para determinar el valor crediticio del solicitante, c) Toma de decisión sobre el crédito. La decisión sobre el crédito a su vez establece si el crédito debe otorgarse y el plan tarifario que puede acceder.

Se requiere de estructuras de redes neuronales para lograr el procesamiento de datos en la verificación de calidad crediticia de las Cuentas aceptadas, las que utilizando el concepto de sistema automático para el control de las Cuentas por Cobrar reducirán la cantidad de tiempo que se usa en éste análisis.

Las mejoras en la calidad de información provocarán que disminuya el riesgo de cada cuenta, al ser ésta una fuente de confiabilidad y seguridad surge de manera natural la rentabilidad a largo plazo, por una reducción de la cartera. Existen muchos factores importantes que deben ser investigados, ya que la calidad de los datos está en los detalles. Estos detalles son parte de la Hoja de Riesgo del Cliente, la cual es un resumen de los datos generales y

económicos del solicitante. Las variables propuestas para que BellSouth tenga una base de datos de calidad son las siguientes:

- Edad
- Estado Civil
- Casa Propia / alquilada
- Trabajo Empleado-Operativo / Ejecutivo-Propietario,
- Años en el último trabajo
- Pago con Débito-Cuenta / Débito-Tarjeta
- Aparece eventualmente en mora por más de un mes

Todos estos datos son tabulados y analizados con el fin de construir un sistema de confiabilidad y así lograr una visión más detallada, crediticia y económica del Cliente y su posible incidencia en las Cuentas por Cobrar.

Después de determinar la estructura procedemos a escoger la regla de aprendizaje de la red y sus parámetros, se presentan otras alternativas de entrenamiento, la factibilidad de perfeccionar las destrezas para pensar y actuar de manera efectiva y realizar una labor eficiente.



# CAPITULO 1

## **1. ANTECEDENTES DE LA CARTERA VENCIDA DE BELL SOUTH, PERÍODO JUNIO DEL 2000 – JUNIO DEL 2001**

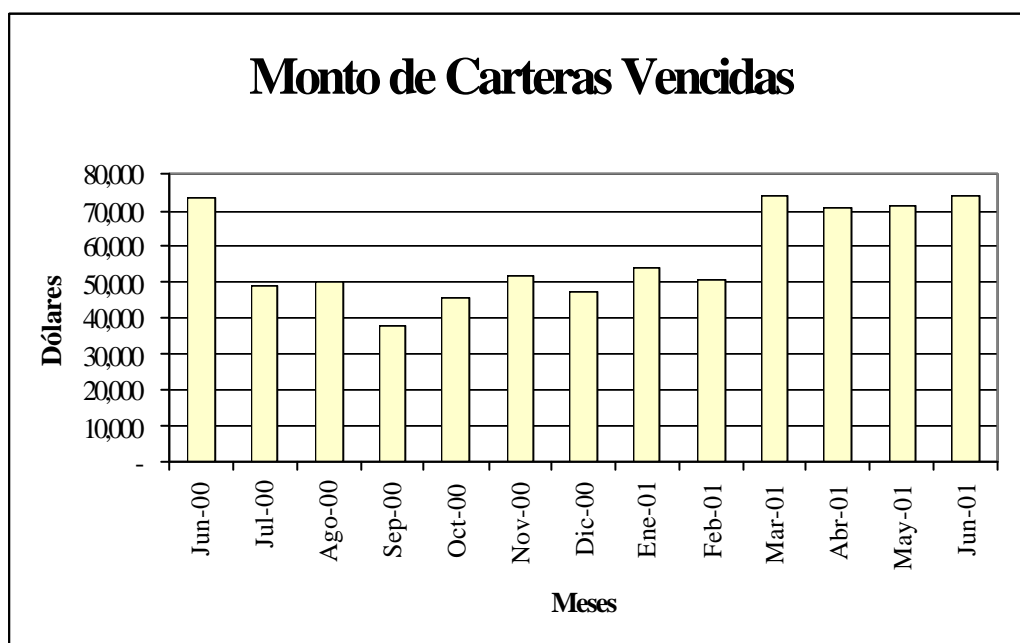
Las compañías operadoras de telefonía celular tienen que vender dos productos: servicio y crédito. En particular nos interesa el producto “crédito” y sólo una fase de aquel. En este período que vive el país, de incertidumbre, los analistas de crédito de BellSouth están tomando decisiones crediticias basándose en el factor de ingresos futuros, antes que el de la liquidez del cliente del servicio.

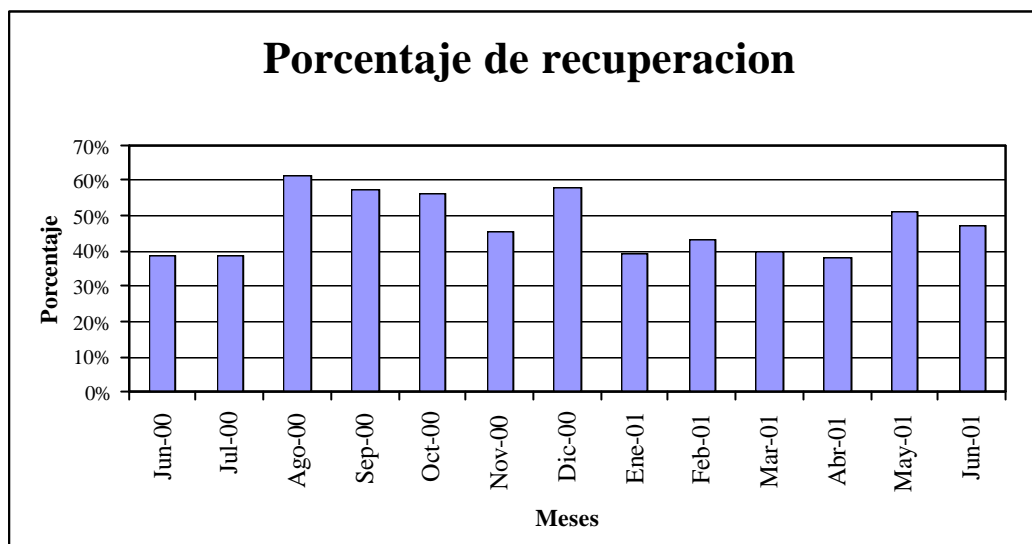
Uno de los factores principales en el análisis del crédito que debe destacarse para poder llegar a una decisión de otorgamiento de crédito es el cliente. El juicio del oficial de créditos y su confianza en dicha persona

pesa fuertemente al tomar la decisión. Existe una reconocida premisa que dice, que un crédito concedido adecuadamente puede considerarse cobrado en un 50%.

En las figuras 1.1 y 1.2 observamos los montos de cartera vencida y los porcentajes de recuperación de cada una en el período de Junio del 2000 a Junio del 2001.

**FIGURA 1.1 Monto de cartera vencida del Período Jun-00 a Jun-01**



**FIGURA 1.2 Porcentaje de Recuperación de la Cartera vencida**

## 1.1. PROCESO DE EVALUACIÓN DE LOS CRÉDITOS INDIVIDUALES DE LA OPERADORA BELLSOUTH GUAYAQUIL ENTREVISTA INICIAL

Todo comienza cuando el probable cliente solicita un plan. El otorgamiento de crédito más prudente se basa en una investigación completa y un análisis competente. El alcance y tipo de riesgo deben ser examinados para determinar la voluntad y habilidad del cliente para pagar su deuda. Son de aplicación los fundamentos básicos de crédito y en ellos ponen atención la operadora. Cada solicitud de crédito requiere un tipo diferente de investigación.

El oficial de créditos debe conducir una entrevista inicial con tacto. El éxito de la entrevista estará garantizado en la medida que el oficial de crédito sepa escuchar, pero debe también hacer preguntas claves, por que el cliente puede omitir consciente o inconscientemente datos importantes. Deberá responder a las siguientes preguntas:

1. ¿Quién es el cliente?
2. ¿Por qué eligió a BellSouth para que le otorgase el servicio de telefonía celular?
3. ¿Qué plan solicita?
4. ¿Cuál es el propósito de solicitar el plan?
5. ¿Cómo resolverá el problema de pago del servicio?
6. ¿De dónde provendrán los fondos para el pago?
7. ¿Qué información financiera dispone el cliente?

Aparte de esta entrevista, existe un formato, el cual debe ser llenado con datos confiables. Seguramente el oficial de créditos necesitará tener algunas referencias adicionales para consultar, el hecho de que desee pagar es tan importante como el hecho de que pueda pagar.

### **INFORMACIÓN**

Buena parte de la información de crédito se obtiene de fuentes externas. Todos los aspectos o vías de información deben ser investigados, sin subestimar nada ni ninguna oportunidad que permita conocer algo más sobre el cliente. Es muy importante que se obtenga información con respecto a la compañía para la cual él trabaja esa combinación de habilidades, buena, mala o indiferente, da lugar al éxito o fracaso del negocio. Esta información no puede ser descartada, y tiene que considerarse como un factor clave de tremendo impacto en la decisión de crédito.

## 1.2. VALORACIÓN CREDITICIA DE LOS CLIENTES DE BELLSOUTH.

Durante la conversación con el cliente las respuestas obtenidas ayudarán a resolver algunas preguntas relacionadas con el carácter, la capacidad, el capital, colaterales y condiciones que rodean tanto al cliente como a la prestación del servicio en sí. La información más trascendental debe ser obtenida por medio del departamento de créditos o el oficial de créditos mediante la investigación.

Las estructuras básicas de las "C" de crédito puede parecer elemental y podría ser sumamente descartada, pero si son pasadas por alto se podría llegar a una decisión que eventualmente resultaría desastrosa para ambos, el cliente y el banco. Examinemos cada "C".

### Carácter

Esta es la más importante de todas las "C". No hay sustituto para el carácter, es un factor vital. El carácter de una persona puede ser causa de que la escala de decisión pueda rotar de "sí" a "no". Honestidad e integridad del cliente son de importancia primordial. La actividad actual y anterior del cliente resulta una buena forma de

opinar para su futura evolución, cuando la compañía está considerando su actual aplicación.

### **Capacidad**

Es la metodología o factor de habilidad gerencial. Explica el éxito de una persona en el transcurso de su vida y cómo lo será posiblemente en el futuro. Un individuo puede manejarse con recursos económicos estables pero si no tiene capacidad y determinación pasará poco tiempo para que se deterioren las condiciones financieras. Carácter y capacidad, ganan el derecho al crédito.

### **Capital**

Este es el respaldo monetario. Indica a BellSouth la habilidad financiera de la persona o cliente. BellSouth necesita conocer de donde proviene esos ingresos o ahorros. El capital es el patrón de medida del crédito que puede otorgarse a aquellos que se han ganado el derecho a solicitar una prestación de servicio de telefonía celular.

### Condiciones

La condición económica del país y la situación de evolución de la empresa para quien trabaja el futuro cliente en el momento de la solicitud, tendrán un peso definitivo sobre la decisión.

Así como consideramos las condiciones económicas, también debemos evaluar las tecnológicas. La velocidad de innovación en el campo técnico puede ser predecesora de acontecimientos económicos que tornarían este crédito indeseable en una fecha futura.

### **INVESTIGACION**

Se trata de la información obtenida de los propios registros de Bellsouth, Bancos, de asociaciones comerciales y principales proveedores de nuestro cliente. Fuentes como informes comerciales, informes de empresas especializadas, nos suministrarán alguna información histórica, así como también información actualizada. Sin embargo, todos ellos tendrán que ser Investigados con profundidad.



Todas las fuentes deben ser investigadas para obtener la información necesaria correspondiente a nuestro cliente, de manera tal, que Bellsouth pueda, a su Juicio, arribar a una decisión adecuada. El oficial de créditos no puede aceptar por sí las cifras de los balances de las cuentas de ahorro - corrientes presentadas por el cliente.

Ya sea solo o acompañado por el asistente de créditos, debe visitar la sede administrativa del cliente, confirmar su domicilio y teléfonos, para lograr visualizar sus activos y las operaciones que él realiza como objetos reales y no meras figuras.

### **¿Qué revela la información bancaria?**

1. Los antecedentes bancarios revelarán en qué manera el cliente ha manejado su cuenta. Esta actuación indicará qué es lo que puede esperarse del cliente en el futuro.
2. Su cuenta corriente indicará si ha habido sobregiros, valores que no hayan sido cobrados, o instrucciones de no pagar cheques.

3. De sus propios registros se podrá extraer información que puede ser beneficiosa acerca de cómo se han manejado las cuentas de sus amigos, parientes y empresas relacionadas.

### **FINALIDAD DEL DEPARTAMENTO DE CREDITOS**

Luego de esta investigación, el oficial de créditos realiza un análisis integral de la evaluación crediticia conducente a una decisión final. El Departamento de Créditos no es un injerto como algunos podrían suponer; es un importante ensamble dentro de la maquinaria administrativa de crédito, cuya finalidad es investigar y analizar toda la información y los estados financieros de cada cliente. Estos son posteriormente recopilados en un informe que se remite al oficial de créditos quien los analiza para poder arribar a una decisión final.

### **1.3. FACTORES DE EVALUACIÓN A LOS CLIENTES DE BELLSOUTH.**

Una vez en nuestro poder la investigación y la información financiera, éstas son seleccionadas para determinar los tres factores previos a la toma de la decisión:

### Personal

Este factor evalúa todas y cada una de las tres 'C' básicas del crédito, haciendo especial énfasis en la honestidad, integridad y capacidad del cliente.

### Económico

Este factor nos suministra las condiciones en que se desenvuelve el cliente, su negocio o empresa para quien trabaja .

### Financiero

Esta es una revisión de todos los datos financieros para que nos aseguren que el cliente está en condiciones de liquidar el crédito.

Esta liquidación puede provenir únicamente de tres fuentes:

1. Transformación de activos en efectivo.
2. Ingresas, ganancias.
3. Tomar préstamo a alguna fuente.

La esencia de la operadora BellSouth es el manejo de liquidez, ella presta servicios de telefonía celular a muchos tipos de clientes durante diferente períodos, en esquemas diferentes, sobre muchas bases y por una variedad de razones. Los créditos, por lo tanto, variarían en liquidez y riesgo.

El propósito de la política crediticia es guiar a los oficiales a, balancear la calidad y cantidad del portafolio de créditos para alcanzar objetivos de ganancia mientras al mismo tiempo se cumplen necesidades crediticias apropiadas, bajo estándares crediticios adecuados, manteniendo el riesgo en límites razonables, minimizando pérdidas, evaluando nuevas oportunidades de negocios y proporcionando adecuada liquidez.

Al buscar tal equilibrio, la calidad de crédito, debe siempre estar primero que el explotar una oportunidad de negocio. Dicho simplemente, la seguridad sobre el servicio y su repago dentro del tiempo establecido tiene prioridad sobre la ganancia en 1a transacción.

#### **1.4. DECISIÓN DE CRÉDITO EN LOS CLIENTES DE BELLSOUTH.**

Los créditos son hechos a individuos, se negocian con gente, por lo tanto, la integridad de cada cliente debe estar por encima de cualquier duda. Las condiciones morales, la integridad y el estilo de hacer negocios deben ser revisados a fondo antes de comenzar negociaciones serias de crédito.

Ahora es el momento de tomar una decisión. Las decisiones de crédito no pueden estar basadas totalmente en cualquier conjunto de lineamiento crediticio o técnica analítica. Cada oficial de crédito debe ejercer sentido común y buen juicio, y ser escrupulosamente prudente. Se espera que el oficial de créditos, tenga en su poder en forma total los detalles cualitativos y cuantitativos de cada transacción, analice sus futuras variables, y prevenga adecuadamente su impacto.

La decisión está basada en un apropiado análisis crediticio de todos los factores mencionados precedentemente y un profundo análisis del balance, de la información financiera obtenida. Tanto el análisis crediticio como el del balance generalmente se llevan a cabo

simultáneamente y uno es complemento del otro. No existe una fórmula o un patrón definido que nos permitan llegar a una decisión final.

Es necesario recordar que antes de entregar cualquier equipo celular deben estar completamente seguros que todos los detalles necesarios hayan sido previstos, tales como garantías colaterales, programa de repago y contrato de servicio pues desde ya nuestro cliente accederá a cualquier cosa para asegurarse que le concedan el crédito.

Pero una vez que éste ha sido concedido ya es demasiado tarde para tratar de protegerse intentando obtener o llegar a un acuerdo sobre estos tópicos.

#### **1.5. LA CARTERA VENCIDA DESDE EL PUNTO DE VISTA DE REDES NEURONALES.**

En el mundo de hoy, donde la tecnología ha adelantado mucho en el campo de la toma de decisión, se han desarrollado varios sistemas

que apretando un botón, obtienen una decisión final a través del procesamiento de datos en el cual están grabados todos los conceptos que he desarrollado en este capítulo, dicho sistema de denomina Redes Neuronales, en el cual no existe un riesgo inherente al factor humano, que es el que juega un papel decisivo en el análisis de todos los hechos.

El análisis de créditos en las redes neuronales es completo, de forma tal, que las condiciones reales, y no las aparentes, del solicitante del crédito pueden ser desentrañadas.

Este análisis involucra un estudio diligente y una comprensión inteligente y viable de las condiciones básicas financieras del cliente en perspectiva; además tiene la capacidad de medir la aptitud o ineptitud natural del cliente para liquidar sus deudas dentro de los plazos pactados.

Recuerden que para ello se utiliza una aproximación algorítmica del reconocimiento de sucesos, porque incrementa de forma

combinatoria el tamaño del espacio admisible de entradas que se va a examinar.

En este sistema se trata de que la red haga la discriminación de características sobre la base de la experiencia previa, determinando así, si un cliente es apto para otorgarle el crédito sin temer que éste se transforme en una cartera rezagada. Estos factores de evaluación para el sistema de redes neuronales ayudan a determinar las ponderaciones para obtener información interna acerca de la naturaleza del problema.

El aspecto más serio, sin embargo es la cantidad de conocimiento que necesita el sistema, no existe una regla explícita para guiarse; es muy similar al aprendizaje de conceptos, se debe introducir una regla a partir de un conjunto de ejemplos de entrenamientos para así obtener una inteligencia artificial y crear nuestro sistema de Redes Neuronales que organice las experiencias, las generalice y utilice para resolver nuevos problemas.



# CAPITULO II

## 2. BREVE DESCRIPCIÓN DEL ANÁLISIS DISCRIMINANTE.

### 2.1 Introducción

La discriminación y la clasificación son técnicas multivariantes concernientes a separar distintos conjuntos de objetos (u observaciones) y con colocar nuevos objetos (observaciones) en grupos previamente definidos. En este capítulo intentaremos básicamente explicar en qué consiste genéricamente la discriminación crediticia que llevaremos a cabo. El análisis discriminante es exploratorio en naturaleza.

Como un procedimiento separativo, a veces es empleado en una base única con el objeto de investigar las diferencias observables cuando las relaciones causales no son bien entendidas. Los

procedimientos clasificatorios son menos exploratorios en el sentido que conllevan hacia reglas bien definidas, que pueden ser usadas para asignar nuevos objetos. La clasificación ordinariamente requiere más estructura del problema que la discriminación.

Entonces, los objetivos inmediatos de la discriminación y clasificación, respectivamente son:

- Describir, sea gráfica (en tres o menos dimensiones) o algebraicamente, las características primarias diferenciales de los objetos (observaciones) de varias colecciones conocidas (poblaciones). Tratamos de encontrar “discriminantes” cuyos valores numéricos sean tales que las colecciones sean separadas tanto como sea posible.
- El sortear objetos (observaciones) en dos o más clases etiquetadas. El énfasis es en derivar una regla que pueda ser usada para asignar óptimamente nuevos objetos hacia las clases predefinidas.

Debemos seguir la convención y usar el término *discriminación* para referirnos al primer objetivo. Esta terminología fue introducida por R. A. Fisher en el primer tratado moderno de problemas separativos. Un término que describe mejor este objetivo es *separación*. Nos referiremos hacia el segundo objetivo como *clasificación* o *colocación*.

Una función que separa objetos puede servir, algunas veces, como un colocador, y, análogamente, una regla que coloca objetos, esto puede sugerir un procedimiento discriminatorio. En la práctica, ambos objetivos se confunden y la distinción entre separación y colocación se torna difusa.

## **2.2 Separación y clasificación para dos poblaciones**

Para fijar las ideas, listemos situaciones en que uno puede estar interesado en (1) separar dos clases de objetos o (2) asignar un nuevo objeto hacia una de dos clases (o ambas). Es conveniente etiquetar las clases como  $\pi_1$  y  $\pi_2$ .

Los objetos son ordinariamente separados o clasificados en la base de mediciones de, por ejemplo,  $p$  variables aleatorias asociadas  $\mathbf{X}' = [X_1, X_2, \dots, X_p]$ .

Los valores observados de  $\mathbf{X}$  difieren en alguna extensión de una clase a otra (sí los valores de  $\mathbf{X}$  no fueran muy diferentes para los objetos en  $\pi_1$  y  $\pi_2$ , no existiría un problema; esto es, las clases serían indistinguibles, y los nuevos objetos podrían ser asignados hacia cualquier clase indiscriminadamente).

Podremos pensar en la totalidad de los valores de la primera clase como ser de la población de  $\mathbf{x}$  valores para  $\pi_1$  y aquellos para la segunda clase como la población de  $\mathbf{x}$  valores para  $\pi_2$ .

Estas dos poblaciones pueden ser entonces descritas por las funciones de densidad de probabilidad  $f_1(\mathbf{x})$  y  $f_2(\mathbf{x})$ , y consecuentemente, podemos hablar de asignar observaciones a poblaciones u objetos a clases indiferentemente.

Para comprender mejor el caso práctico la siguiente lista nos ilustra algunos ejemplos de este planteamiento.

TABLA 2.1 Ejemplos de Clasificación y Discriminación

| <b><i>Poblaciones <math>p_1</math> y <math>p_2</math></i></b>  | <b><i>Variables Medidas <math>X</math></i></b>  |
|--|---|
| 1. Compañías de seguros solventes y financieramente estresadas   | Activos totales, costo de stocks y bonos, valor del stock en el mercado, gastos de pérdidas, ahorros, cantidad de primas contratadas. |
| 2. Discépticos Ulcéricos (con problemas de estómago sensible) y controles ("normales")                   | Medidas de ansiedad, dependencia, culpa, perfeccionismo   |
| 3. <i>Documentos Federalistas</i> escritos por James Madison y aquellos escritos por Alexander Hamilton. | Frecuencia de distintas palabras y largo de las oraciones.  |
| 4. Dos especies de hierba  | Largo de sépalos y pétalos, profundidad de las grietas de los pétalos, largo de la hoja, diámetro del polen.                          |

|  |   |
|--|---|
| 5. Compradores de un nuevo producto y aquellos compradores que dubitan al comprar. | Educación, ingreso, tamaño de la familia, cantidad de marcas previamente cambiadas.   |
| 6. Estudiantes universitarios exitosos y no exitosos (fallaron al graduarse)       | Notas del examen de ingreso, nota promedio de graduación del colegio, número de actividades extracurriculares de la secundaria. |
| 7. Hombres y mujeres   | Medidas antropológicas tales como circunferencia y volumen de cráneos antiguos.   |
| 8. Buenos y malos sujetos de crédito*  | Ingreso, edad, número de tarjetas de crédito, tamaño de la familia.   |
| 9. Alcohólicos y no alcohólicos  | Actividad de las enzimas de monoaminas oxidadas, actividad de las enzimas de adenilato ciclosa.                                 |

Vemos del ítem 5, por ejemplo, que los objetos (consumidores) serían separados en dos clases etiquetadas (“compradores” y “no compradores”) en la base de valores observados de presumiblemente variables relevantes (educación, ingreso, etc.). En la terminología de observación y población, queremos identificar una observación de la forma  $x' = [x_1 \text{ (educación), } x_2 \text{ (ingreso), } x_3 \text{ (tamaño familiar), } x_4 \text{ (número de cambios de marca)]$  como una población  $\pi_1$ , compradores, o población  $\pi_2$ , no compradores.

### **2.2.1 Clasificación de dos poblaciones**

La colocación o clasificación tiene reglas que son usualmente desarrolladas de ejemplos de entrenamiento (tal y como ocurre en las redes neuronales).

Las características medidas de objetos, aleatoriamente elegidos sabiendo que provienen de cada una de las dos poblaciones son examinadas para distinguir sus diferencias.

Esencialmente, el conjunto de todos los posibles resultados se divide en dos regiones,  $R_1$  y  $R_2$ , tal que si una nueva observación cae en  $R_1$ , se coloca en la población  $\pi_1$ , y si cae en  $R_2$ , la colocamos en  $\pi_2$ . Por tanto, un conjunto de observaciones favorece a  $\pi_1$ , mientras que el otro conjunto de valores favorece a  $\pi_2$ .

Talvez nos estemos preguntando cómo es que sabemos que algunas observaciones pertenecen a una población en particular, pero dudamos de otras. (Esto, por supuesto, es lo que hace a la clasificación un problema) Algunas condiciones pueden originar esta aparente anomalía:

- *Conocimiento incompleto de un funcionamiento futuro.*  
*Ejemplo:* En el pasado los valores extremos de variables financieras eran observados con dos años de anterioridad antes de que ocurra la subsecuente bancarrota de alguna firma. Clasificar otra firma como “estresada” sobre la base de valores observados de estos indicadores liderantes, podrían permitir a los oficiales, el tomar acciones



correctivas, de ser necesario, antes de que sea demasiado tarde. Otro ejemplo; una oficina de aplicaciones para un colegio médico quisiera clasificar un aplicante como un posible o no posible candidato a convertirse en Médico sobre la base de las notas de los exámenes y otros registros universitarios. Aquí, una determinación actual puede ser hecha únicamente al final de varios años de entrenamiento.

- *Información “perfecta” requiere destruir el objeto. Ejemplo:* La vida útil de una batería de calculadora se determina al usarla hasta que falle, y la resistencia de un trozo de madera se obtiene al doblarlo hasta que se quiebra. Los productos fallosos no pueden ser vendidos. Uno quisiera clasificar productos como buenos y malos (que no cumplen con las especificaciones) sobre la base de ciertas mediciones preliminares.
- *Información no disponible o costosa. Ejemplo:* Se asume que algunos de los *Documentos Federalistas* fueron

escritos por James Madison o Alexander Hamilton porque ellos los firmaron. Otros documentos, sin embargo, no fueron firmados y es de interés el determinar cuál de los dos hombre escribieron los papeles sin firmar. Claramente, no podemos preguntarles. Las frecuencias de las palabras y largo de las oraciones podrían ayudar a clasificar los papeles en disputa. Otro ejemplo; muchos de los problemas médicos pueden ser identificados concluyentemente sólo al conducir una costosa operación. Usualmente, uno quisiera diagnosticar una enfermedad a través de fácilmente observables, aunque potencialmente fallidos, síntomas externos. Esta aproximación ayuda a evitar operaciones costosas e innecesarias.

Debe estar claro de estos ejemplos que las reglas de clasificación no pueden usualmente proveer un método de asignación libre de errores.

Esto ocurre porque puede no existir una clara distinción entre las características medidas de las poblaciones; esto es, los

grupos pueden traslaparse. Entonces es posible, por ejemplo, el clasificar incorrectamente a un objeto perteneciente a  $\pi_1$  en  $\pi_2$  o perteneciente a  $\pi_2$  en  $\pi_1$ .

Consideremos el siguiente ejemplo: (Discriminando poseedores de no poseedores de cortadoras de césped)

Consideremos dos grupos en una ciudad:  $\pi_1$ , los dueños de una cortadora de césped, y  $\pi_2$ , aquellos que no tienen una.

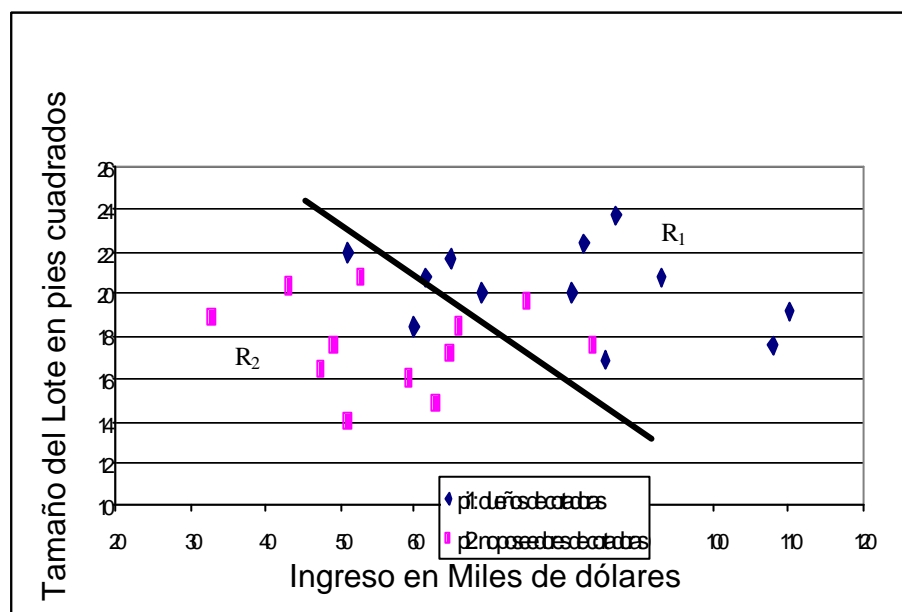
Para poder identificar los mejores prospectos de ventas para una intensa campaña de ventas, un fabricante de cortadoras de césped está interesado en clasificar a las familias como prototipo de poseedores y no poseedores sobre la base de  $x_1$  = ingreso y  $x_2$  = tamaño del lote. Muestras aleatorias de  $n_1 = 12$  dueños actuales y  $n_2 = 12$  no poseedores actuales produjeron los valores en la tabla 2.2.

Tabla 2.2 Clasificación de Familias

| <i>p</i> <sub>1</sub> : <b>dueños de cortadoras</b> |                                      | <i>p</i> <sub>2</sub> : <b>no poseedores de cortadoras</b> |                                      |
|---|--------------------------------------|--|--------------------------------------|
| x1 ( ingreso en miles \$)                           | x2 (área del lote en pies cuadrados) | x1 ( ingreso en miles \$)                                  | x2 (área del lote en pies cuadrados) |
| 60  | 18.4                                 | 75   | 19.6                                 |
| 85.5  | 16.8                                 | 52.8   | 20.8                                 |
| 64.8  | 21.6                                 | 64.8   | 17.2                                 |
| 61.5  | 20.8                                 | 43.2   | 20.4                                 |
| 87  | 23.6                                 | 84   | 17.6                                 |
| 110.1   | 19.2                                 | 49.2   | 17.6                                 |
| 108   | 17.6                                 | 59.4   | 16                                   |
| 82.8  | 22.4                                 | 66   | 18.4                                 |
| 69  | 20                                   | 47.4   | 16.4                                 |
| 93  | 20.8                                 | 33   | 18.8                                 |
| 51  | 22                                   | 51   | 14                                   |
| 81  | 20                                   | 63   | 14.8                                 |

Estos datos se muestran en la figura 2.1. Vemos que los dueños de estas cortadoras tienden a tener mayores ingresos y más grandes lotes que los que no tienen cortadoras, aunque el ingreso parece ser un mejor “discriminador” que el tamaño del lote.

**Figura 2.1 Ingreso y tamaño de lote para poseedores y no poseedores de cortadoras de césped**



Por otra parte, existe algún traslapamiento entre ambos grupos. Si, por ejemplo, tuviéramos que colocar aquellos valores de  $(x_1, x_2)$  que caen en la región  $R_1$  (como lo determina la línea negreada de la figura) en  $\pi_1$ , poseedores, y aquellos valores  $(x_1, x_2)$  que caen en  $R_2$  para  $\pi_2$ , no poseedores, cometeríamos errores. Algunos dueños serían clasificados incorrectamente como no poseedores e inversamente algunos no poseedores como poseedores. La idea es crear una regla en las regiones  $R_1$  y  $R_2$  que minimice los chances de cometer errores.

Un buen procedimiento clasificatorio debe resultar en pocas colocaciones fallidas. En otras palabras, los chances, o probabilidades, de malas clasificaciones deberían ser pequeñas. Cabe mencionar que existen características adicionales que una regla de clasificación “óptima” debe tener.

Podría ocurrir que una clase o población tenga una mayor propensión de ocurrencia que otra debido a que una o dos poblaciones sean relativamente mucho más largas que otras.

Por ejemplo, tiende a haber más firmas financieramente estables de las que no lo están. Como otro ejemplo, una especie de hierba puede ser más prevaleciente que otro. Una regla de clasificación óptima debería tomar estas “probabilidades previas de ocurrencia” en cuenta.

Si realmente creemos que la probabilidad (a priori) de una firma financieramente estresada y finalmente en bancarrota es muy pequeña, entonces debemos clasificar una firma elegida al azar como no en bancarrota a no ser que los datos eminentemente favorezcan a la bancarrota.

Otro aspecto de la clasificación es el costo. Supongamos que clasificar un objeto  $\pi_1$  como pertenecedor a  $\pi_2$  representa un error mucho más serio que clasificar un objeto  $\pi_2$  como pertenecedor a  $\pi_1$ . Entonces debemos ser cautelosos al hacer futuras asignaciones.

Como un ejemplo, el fallar al diagnosticar una enfermedad potencialmente fatal es substancialmente más “costoso” que concluir que la enfermedad está presente cuando, de hecho, no lo está. Un procedimiento clasificatorio debería, en la medida de lo posible, contar con los costos asociados a las malas clasificaciones.

## **2.3 Observaciones**

### **2.3.1 Incluyendo variables cualitativas**

Hasta ahora la discusión ha sido en torno a que las variables discriminatorias  $X_1, X_2, \dots, X_P$  tengan unidades naturales para su medición.

Esto es, cada variable puede, en principio, asumir cualquier número real, y estos números pueden ser registrados. A veces, una variable cualitativa o categórica puede ser un discriminador útil (clasificador).



Por ejemplo, la presencia o ausencia de una característica tal como el color rojo puede ser un clasificador justificadamente útil. Esta situación se maneja frecuentemente al crear una variable  $X$  cuyos valores numéricos sean uno (1) si el objeto posee la característica y cero (0) si el objeto no posee la característica.

La tabla es luego tratada como las variables medidas usualmente en el proceso discriminatorio – clasificadorio.

Hay muy poca teoría disponible para manejar el caso en que algunas variables son continuas y algunas otras cualitativas. Los experimentos de simulación computacional indican que la función lineal de clasificación de Fisher [ver bibliografía] puede responder pobre o satisfactoriamente, dependiendo de las correlaciones entre las variables cualitativas y continuas.

Cuando un número de variables son del tipo 0 – 1, podría ser mejor el considerar una aproximación mejor, llamada

aproximación de regresión logística hacia la clasificación. La probabilidad de pertenencia hacia el primer grupo,  $p_1(x)$ , se modela directamente como:

$$p_1(x) = \frac{e^{a+b'x}}{1 + e^{a+b'x}}$$

Para el problema de dos poblaciones. El  $\alpha$  necesita ser ajustado para acomodar una distribución a priori, y puede no ser fácil el incluir costos. Si las poblaciones son cercanamente normales con matriz de covariancias iguales, la aproximación lineal de clasificación es mejor.

En el siguiente capítulo se notará cuán coherente es la utilización de esta función logística o sigmoidea para la tarea discriminatoria – clasificatoria.

### **2.3.2 Observaciones finales**

Hemos querido explicar el problema de clasificación desde la óptica genérica del análisis discriminante. Aunque esta tesis no incluye un análisis discriminante tradicional, si es válido

decir que el proceso clasificatorio que lleva a cabo una red neuronal tiene naturaleza discriminante y la tarea o fin, es exactamente el mismo, aunque los procedimientos sean distintos.

Debido a esto, no se incluye más que una explicación somera del análisis discriminante en este capítulo ya que, entrar en materia más profunda escapa al alcance de esta tesis.

En esta investigación nos hemos dado cuenta que la función sigmoide tiene un uso común, eso lo corroboramos en el tercer capítulo.

# CAPITULO III

## 3. REDES NEURONALES ARTIFICIALES

### 3.1 Introducción

La mente humana surge como modelo para máquinas inteligentes ya que en principio es una obvia idea el imitar su comportamiento. Una simulación en computadora del funcionamiento del cerebro ha sido un objeto de estudio de la IA (Inteligencia Artificial) desde los años cuarenta del siglo pasado. Las poderosas cualidades de la mente en lo que respecta a pensamiento, recordación y solución de problemas, a inspirado a los científicos el intentar el modelamiento computarizado de su operación. Un grupo de investigadores se propuso crear un modelo computacional que coincidiera con la funcionalidad de la mente de una fundamental manera y el resultado ha sido la neurocomputación.

## **3.2 La neurocomputación**

### **3.2.1 Analogía con el cerebro**

La neurona es la unidad celular fundamental del sistema nervioso y, en particular, del cerebro. Cada neurona es una unidad sencilla de microprocesamiento que recibe y combina señales desde muchas otras neuronas a través de estructuras de procesos de entrada llamadas dendritas.

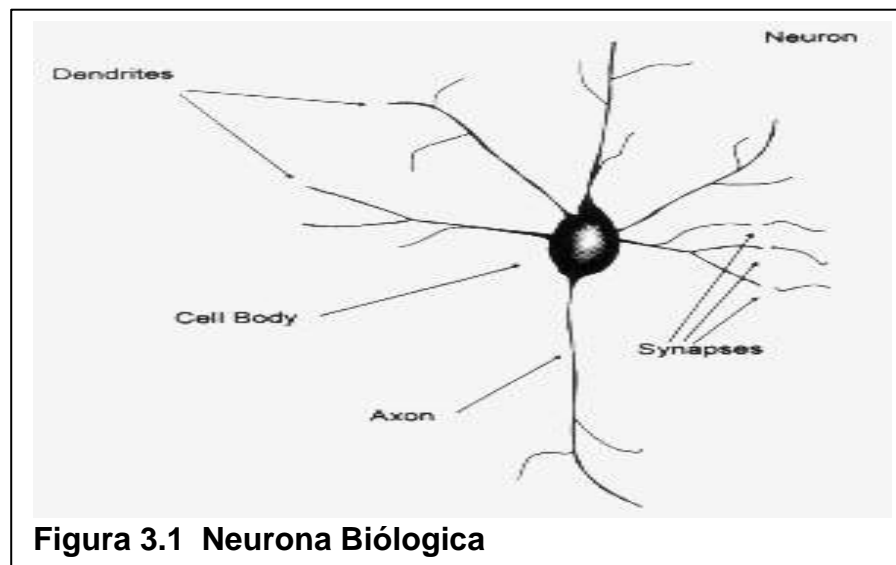
Si la señal combinada es lo suficientemente fuerte, activa el disparo de la neurona, lo que produce una señal de salida; la ruta de la señal de salida se da sobre un componente llamado axón. Esta simple transferencia de información tiene naturaleza química, pero tiene efectos colaterales de tipo eléctrico que podemos medir.

Se estima que el cerebro consiste de algunas decenas de miles de millones de neuronas densamente interconectadas. El axón (ruta de salida de la señal) de una neurona se divide y se conecta hacia las dendritas (ruta de entrada) de otra neuronas mediante una unión llamada sinápsis. La

transmisión sobre esta unión es química en naturaleza y la cantidad de señal trasferida depende de la cantidad de químicos (llamados neurotransmisores) liberados por el axón y recibidos por las dendritas.

Esta eficiencia sináptica (o resistencia) es la que es modificada cuando la mente aprende. La sinápsis combinada con el procesamiento de información en la neurona forman el mecanismo básico de memoria en el cerebro.

La figura 3.1 ilustra una neurona biológica.



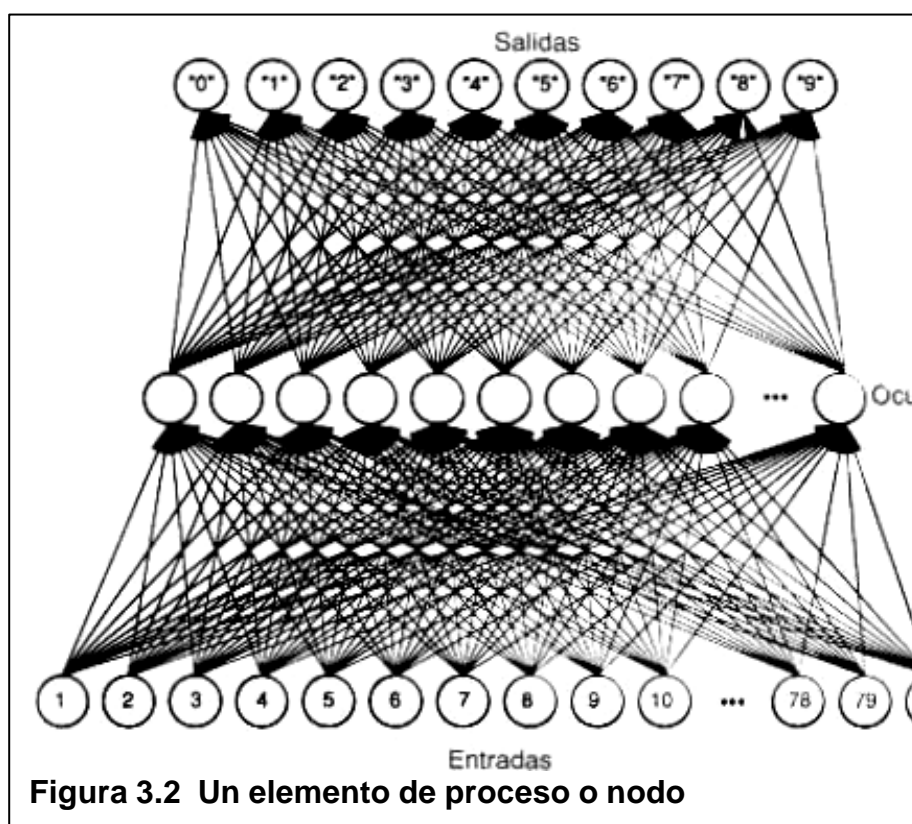
**Figura 3.1 Neurona Biológica**

### 3.2.2 Redes Neuronales

En una red neuronal artificial la unidad análoga a la neurona biológica es referida como un “nodo” o “elemento de procesamiento”. Un nodo tiene muchas entradas (dendritas) y combina, usualmente a través de una suma, los valores de estas entradas. El resultado es un nivel de actividad interna para el nodo. Las entradas combinadas son luego modificadas por una función de transferencia. Esta función de transferencia puede ser de tipo umbral lo que hará, que sólo pase información si el nivel de actividad combinado llega a un cierto nivel, o puede ser una función continua de la combinación de las entradas. El valor de salida de la función de transferencia es generalmente pasado directamente hacia la ruta de salida del nodo.

La ruta de salida de un nodo puede ser conectada a entradas de otros nodos por medio de ponderaciones que corresponden (análogamente) a la resistencia sináptica de las conexiones neuronales. Como cada conexión posee una correspondiente ponderación o peso, las señales de las líneas de entrada hacia

un nodo son modificadas por estos pesos previamente antes de ser sumadas. Es decir, la función de suma es una sumatoria ponderada. En si mismo, este modelo simplificado de una neurona no es muy interesante; los efectos interesantes resultan de las maneras en que las neuronas sean interconectadas.



Una red neuronal consiste de muchos nodos unidos o conectados de la manera en que se muestra en la figura 3.2. Los nodos son usualmente organizados en grupos llamados



“capas”. Una red típica consiste de una secuencia de capas con total o aleatorias conexiones entre capas sucesivas.

Existen usualmente dos capas con conexiones hacia el mundo exterior: un búfer de entrada donde los datos se le presentan a la red, y un búfer de salida que contiene la respuesta de la red a una entrada dada. Las capas intermedias se las denomina “capas ocultas”.

### **3.2.3 Operación de las redes**

Hay dos fases principales en la operación de una red – Aprender y Recordar. En la mayoría de las redes esto es distinto.

“Aprendizaje” es el proceso de adaptación o modificación de los pesos en respuesta hacia estímulos presentados en la capa de entrada y opcionalmente la capa de salida. Un estímulo presentado en la capa de salida corresponde hacia una respuesta deseada debido a una entrada dada; esta

respuesta deseada debe ser provista por un “instructor con conocimiento”. En tal caso el aprendizaje es referido como “aprendizaje supervisado”.

Si la salida deseada es diferente de la entrada, la red entrenada es conocida como “red hetero-asociativa”. Si, para todo ejemplo de entrenamiento, el vector de salida requerido es igual al vector de entrada, la red entrenada se la llama “auto-asociativa”. Si no se muestra ninguna salida requerida, el aprendizaje se denomina “no supervisado”.

Un tercer tipo de aprendizaje, que cae entre supervisado y no supervisado, es el “aprendizaje de reforzamiento” donde un instructor externo indica si es que la respuesta es buena o mala. En algunas instancias, la red puede ser calificada sólo después de que algunas entradas han sido procesadas por la red.

Cualesquiera tipo de aprendizaje que sea usado, una esencia característica de toda red es su “regla de aprendizaje”. La regla de aprendizaje especifica cómo los pesos se adaptan en respuesta a un ejemplo de entrenamiento. El aprendizaje puede requerir mostrar a la red muchos ejemplos, muchas miles de veces, o solamente una vez.

Los parámetros que gobiernan la regla de aprendizaje pueden cambiar a través del tiempo mientras que la red progresa en su aprendizaje. El control a largo plazo de los parámetros de aprendizaje es conocido como un calendario de aprendizaje.

“Recordar” se refiere a cómo la red procesa un estímulo presentado en su búfer de entrada y crea una respuesta en su búfer de salida. A menudo la recordación es una parte integral del proceso de aprendizaje así como cuando una respuesta deseada de la red deba ser comparada con el valor real de la salida para crear una señal de error.

La forma más sencilla de red neuronal no tiene conexiones de retroalimentación de una capa hacia otra o hacia sí misma. Tal red se denomina “feedforward network” (o algo así como red de pro-alimentación).

Es este caso, la información pasa del búfer de entrada, hacia capas intermedias y hacia la capa de salida, de manera progresiva, usando las características de las funciones de suma ponderada y de transferencia de la particular red en cuestión.

Las Feedforward networks son interesantes debido a no-linealidades en las transformaciones. En algunas de estas redes, cierta cantidad de retroalimentación se crea para que la red tenga sensibilidad al tiempo, es decir, tener puntos de referencia temporales. Estas se llaman redes de recurrencia o de retroalimentación.

Si hay conexiones de retroalimentado, la información reverberará alrededor de la red, sobre la capas o dentro de las mismas, hasta que exista algún criterio de convergencia que sea alcanzado. La información es luego pasada hacia la capa de salida.

### **3.2.3.1 Operación de las capas**

Dos operaciones se usan en la neurocomputación que pueden afectar una red totalmente como un todo, estas son la Normalización y la Competición.

- Normalización: toma el vector de valores, correspondiente a la salida de una capa completa, y la escala para que con esto la salida total (por ejemplo la suma de los componentes del vector) sea un valor fijado. Esto se hace en sistemas biológicos al conectar cada elemento de una capa a cada otro elemento. Estas conexiones permiten al nodo que sienta individualmente la salida total de la capa y ajuste sus propios valores

concordantemente. El resultado de la normalización es que la total actividad en la capa permanece aproximadamente constante.

- **Competición:** se refiere a la interacción que un nodo puede tener con cada otro nodo en la misma capa. A diferencia de la normalización, donde todos los nodos ajustan sus salidas para crear un nivel fijado de actividad, en la competición sólo una o unas pocas neuronas ganan y producen una salida. Una forma común de competencia es cuando el nodo con mayor actividad es el único en su nivel que se dispara (que significa que da salida a su actual estado).

### **3.2.4 Qué hace a la neurocomputación diferente?**

La neurocomputación difiere de la inteligencia artificial común y la computación tradicional en algunas formas importantes.

### **3.2.4.1 Aprendizaje a través de ejemplos**

A diferencia de los tradicionales sistemas expertos donde el conocimiento se hace explícito a través de reglas, las redes neuronales generan sus propias reglas al aprender de ejemplos mostrados hacia ellas. El aprendizaje se logra a partir de una regla de entrenamiento que adapta o modifica los pesos de conexión de la red en respuesta de entradas de ejemplos y (opcionalmente) las deseadas salidas de aquellas entradas.

En el aprendizaje no-supervisado, se muestran a la red sólo los estímulos de entrada y la red por sí misma se organiza internamente para que cada nodo oculto responda fuertemente hacia un conjunto diferente de estímulos de entrada o a un conjunto de estímulos cercanamente relacionado. Estos conjuntos de estímulos de entrada representan grupos en el espacio de entrada que típicamente representan distintos conceptos del mundo real.

En el aprendizaje supervisado, por cada estímulo de entrada, un estímulo de salida deseado se presenta al sistema y la red se configura a sí misma gradualmente para alcanzar esa correspondencia deseada de entrada / salida. Tal aprendizaje es generalmente una variante de alguno de estos tres tipos:

- Aprendizaje Hebbiano: donde un peso de conexión de una ruta de entrada hacia un nodo es incrementado si la entrada es de alto nivel y si la salida deseada también. En términos biológicos, esto quiere decir que una ruta neuronal se ve reforzada cada vez que se es activada la sinápsis que correlaciona ambas neuronas.
- Aprendizaje de la Regla Delta: se basa en reducir el error entre la salida actual de un nodo y la salida deseada al modificar las ponderaciones. Esta es la regla que usaremos en nuestro caso; cabe notar



que este concepto se asemeja mucho al proceso de mínimos cuadrados en la regresión lineal.

- **Aprendizaje Competitivo:** en el que los elementos de procesamiento compiten entre ellos, y aquel que obtenga la respuesta más fuerte hacia una entrada dada se modifica a sí mismo para tratar de tornarse en aquella entrada.

#### **3.2.4.2 Memoria de Asociación Distribuida**

Una característica importante de las redes neuronales es la forma en que guardan información. La memoria neurocomputacional es distribuida. Los pesos de conexión son las unidades de memoria de una red neuronal. Los valores de las ponderaciones representan el actual estado de conocimiento de la red. Un elemento de conocimiento, representado, por ejemplo, por un par de entrada / salida deseado, se distribuye a través muchas de las unidades de memoria en la red y comparte estas unidades de

memoria con otros elementos de conocimiento guardados en la red.

Ciertas memorias neurocomputacionales son asociativas en que, si la red entrenada se presenta con una entrada parcial, la red escogerá la opción más cercana en memoria para dicha entrada, y generará una salida que corresponderá a una entrada completa. Si la red es auto-asociativa (que es, la entrada es igual a la salida deseada para todos los pares de ejemplos usados para entrenar la red), el presentarle a la red vectores parciales de entrada resultarán en su completación.

La naturaleza de la memoria de una red neuronal conlleva hacia respuestas razonables al presentársele entradas incompletas, ruidosas, o jamás vistas. La propiedad se la denomina “generalización”. La calidad y significancia de una generalización depende de la aplicación particular, y en el tipo y la sofisticación de la

red. Redes multicapa no-lineales (en particular, redes “back-propagation”) aprenden acerca de características en sus capas ocultas y combinan esto para producir las salidas. El conocimiento en las capas ocultas puede combinarse para formar respuestas “inteligentes” a estímulos nuevos.

#### **3.2.4.3 Tolerancia a las fallas**

Mientras que los sistemas tradicionales de computación se vuelven inútiles por hasta una pequeña cantidad de daño en la memoria, los sistemas de neurocomputación son tolerantes ante las fallas. La tolerancia a las fallas se refiere a el hecho que en la mayoría de las redes neuronales, si algunos nodos son destruidos, truncados, desactivados, o sus conexiones pequeñamente alteradas, entonces el comportamiento de la red se degrada apenas un poco más.

El desempeño sufre, pero el sistema no es llevado a tan abrupta falla. Los sistemas neurocomputacionales son tolerantes ante las fallas porque la información no está contenida en un solo lugar, pero está distribuida a lo largo del sistema.

Esta característica de degradación “agraciada” hace a los sistemas neurocomputacionales extremadamente convenientes para aquellas aplicaciones en que una falla en equipos de control significa desastre, por ejemplo, en una planta de energía nuclear, guía de misiles, misiones espaciales, etc.

#### **3.2.4.4 Reconocimiento de patrones**

Los sistemas neurocomputacionales son adeptos a muchas tareas de reconocimiento de patrones, aún mucho más que los tradicionales sistemas expertos o estadísticos.

La habilidad humana de traducir los símbolos de esta página hacia palabras con significado e ideas es una forma de reconocimiento de patrones.

Las tareas de reconocimiento de patrones requieren la habilidad de coincidir grandes cantidades de información de entrada simultáneamente, y luego generar salidas generalizadas o categorizadas. También requieren respuestas razonables hacia entradas ruidosas.

Los sistemas neurocomputacionales poseen estas capacidades así mismo como la habilidad de aprender y construir estructuras únicas para problemas específicos o particulares, por lo tanto son especialmente útiles en reconocimiento de patrones.

La habilidad de seleccionar combinaciones de características, pertinentes al problema, les dan una

ventaja por encima de los sistemas basados en la estadística. La habilidad de deducir estas características por sí mismas les da una ventaja por sobre los sistemas expertos utilizados en clasificación de patrones. Es evidente que esta es la principal razón por la que una red neuronal puede servir para clasificar el perfil de un cliente en un análisis crediticio.

#### **3.2.4.5 Poder de Síntesis**

Ciertas redes neuronales son capaces de aprender mapeos (correspondencias funcionales) complejos y continuos de una o más entradas hacia una o más salidas. Esta habilidad para sintetizar funciones continuas complejas es análoga a la habilidad de los sistemas biológicos para aprender movimientos coordinados, por ejemplo mover una pierna y pegarle a una pelota.

### 3.2.5 El Perceptrón de Frank Rosenblatt

En los distintos textos sobre el tema existen varios investigadores que siempre son mencionados por sus trabajos al respecto, tales como McCulloch y Pitts, Minsky, Papert, Block, y el más famoso Rosenblatt.

En 1957, Frank Rosenblatt, en Cornell, publicó el primer mayor proyecto de investigación en neurocomputación: el desarrollo de un elemento llamado “perceptrón”. El perceptrón de Rosenblatt provocó una gran cantidad de interés investigativo en la neurocomputación.

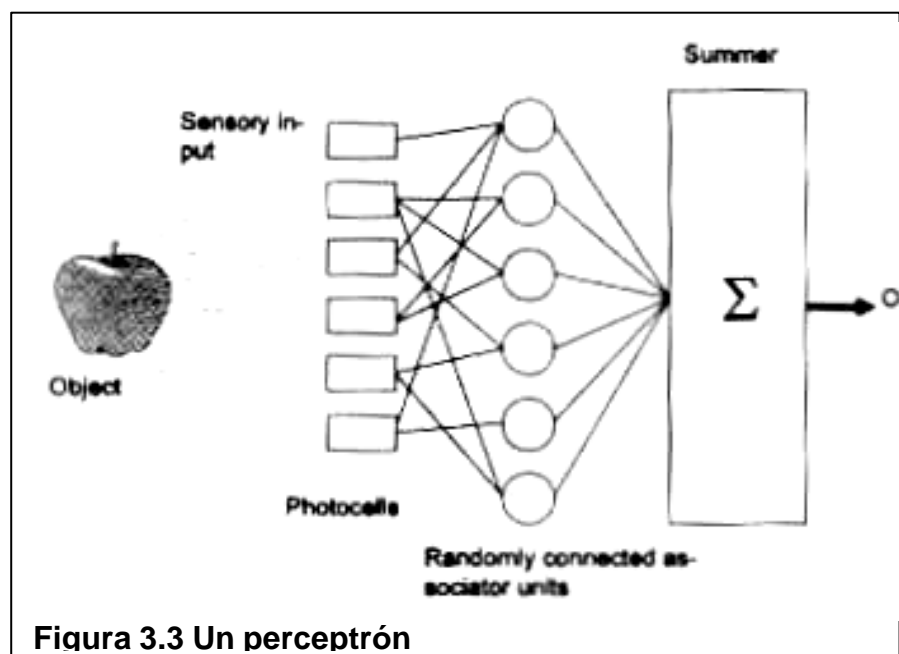


Figura 3.3 Un perceptrón

El perceptrón es un sistema de clasificación de patrones que puede identificar ambos, abstractos y geométricos. El primer perceptrón era capaz de algún aprendizaje y era robusto con respecto a que su operación era únicamente degradada después de dañarse partes de sus componentes.

El perceptrón también poseía una gran cantidad de plasticidad, significando esto que puede ser restringida a un punto en que no haya errores luego que algunas células sean destruidas. Además, el perceptrón era capaz de hacer generalizaciones limitadas y podía categorizar propiamente patrones a pesar de ruido existente en las entradas.

El perceptrón fue primariamente apuntado hacia el reconocimiento óptico de patrones. Una rejilla de 400 fotocélulas, correspondientes a neuronas sensibles a la luz en la retina, recibían el estímulo óptico primario. Estas fotocélulas estaban conectadas por unidades asociadoras que recogían los impulsos eléctricos de las fotocélulas.

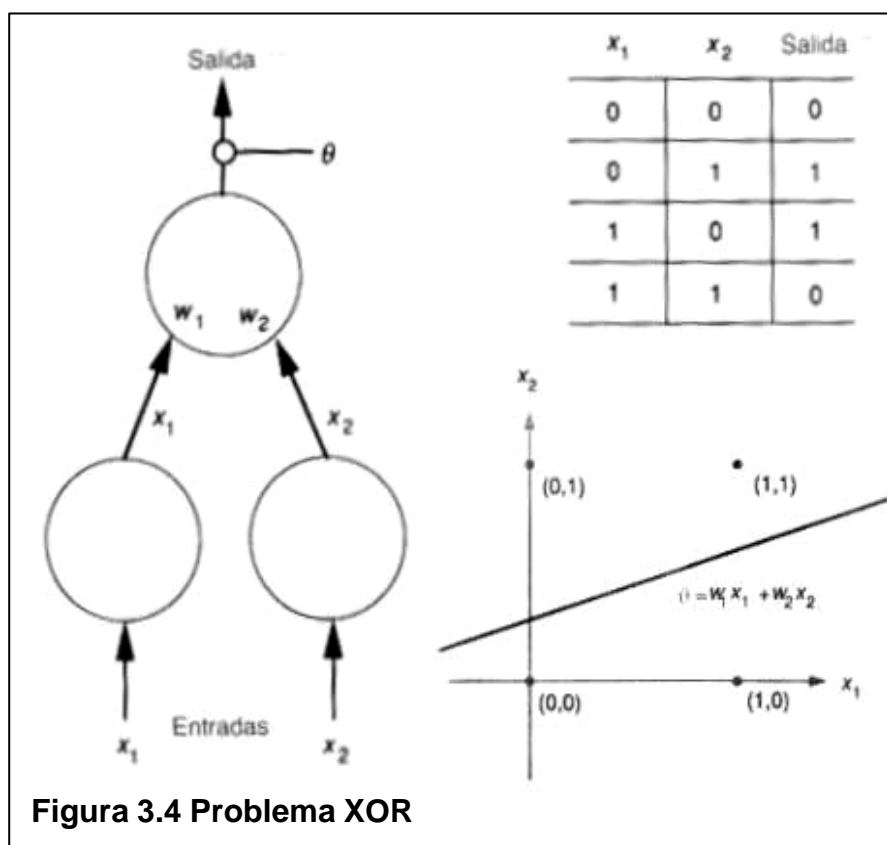


Las conexiones entre las unidades asociadoras que recogían los impulsos eléctricos eran hechas al cablear aleatoriamente los asociadores hacia las células. Si la entrada de las células excedía cierto umbral, las unidades de asociación provocaban una respuesta para provocar una salida.

Debido a que era un aparato en desarrollo, el perceptrón también tenía ciertas limitaciones. El perceptrón no tenía estructuras de estados y era inapropiado para descripciones en términos de la teoría de los autómatas. Otra gran limitación, que sería enfatizada por Minsky y Papert, era la inhabilidad del perceptrón para representar el problema básico de la función  $\oplus$  Exclusiva (XOR). Por supuesto esto es el resultado de la naturaleza lineal del perceptrón.

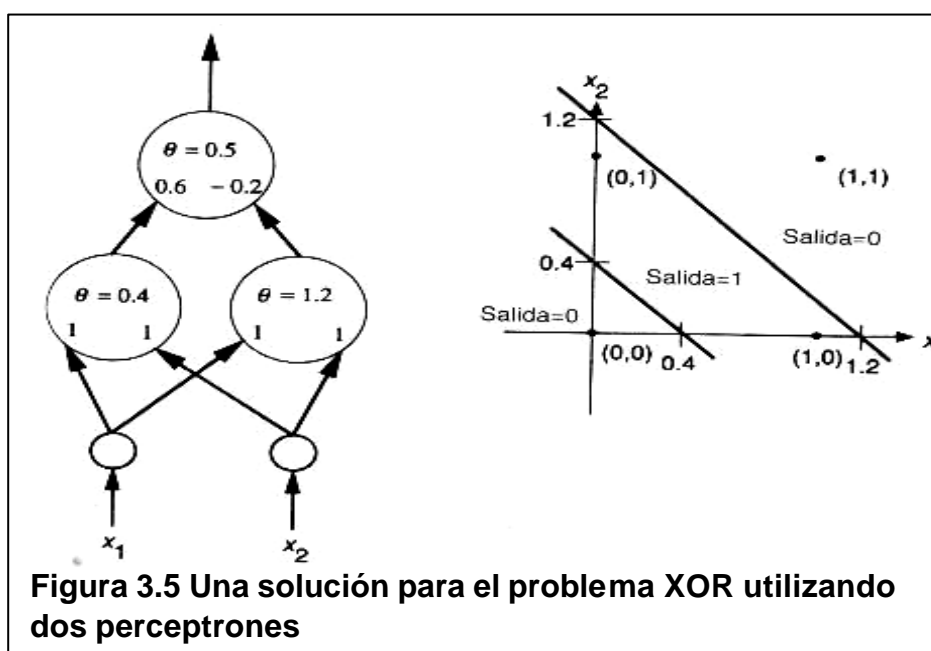
Uno de los cambios más significativos hechos en los años sesentas después de Rosenblatt ha sido el desarrollo de sistemas multicapas, que pueden aprender y categorizar estructuras complejas y no lineales. Esto se hace típicamente

al usar una función de transferencia no lineal, pero puede también surgir de la normalización y competición.



Es fácil entender por que un perceptrón no puede resolver problemas de separabilidad lineal. El problema XOR se ilustra en la figura 3.4. No existe perceptrón alguno que sea capaz de discriminar entre ambos grupos.

Esto se resuelve al utilizar un conjunto de perceptrones (una red) por ejemplo, en la figura 3.5. se da una solución al problema XOR utilizando dos perceptrones. Así, una red neuronal primitiva, como lo es un perceptrón, se puede tornar en una herramienta que permita discriminar patrones aunque estos no tengan características lineales como XOR.



No existe línea recta capaz de separar ambos grupos, pero si existe una función no-lineal capaz de hacerlo. Una red multicapa puede emular este comportamiento y esa es una cualidad que aprovecharemos para hacer la discriminación para el análisis crediticio de clientes de Bellsouth. También es por esta razón que en un previo capítulo se describió brevemente lo que significaba un análisis discriminante.

### 3.2.6 Propagación hacia atrás (Back-Propagation)

“Back-propagation” es una técnica para resolver el problema de asignamiento de crédito mencionado por Minsky y Papert en su libro *Perceptrons*. David Rumelhart, es una persona asociada con la invención de las redes de back-propagation. David Parker introdujo un algoritmo similar casi al mismo tiempo y otros han estudiado redes similares.

Una red de perceptrones es capaz de entrenar los nodos de salida para aprender a clasificar patrones de entrada, dado que las clases son “linealmente separables”. Las clases más complejas no-linealmente separables pueden ser separadas con una red multicapa. De todas formas, si la salida esta errada, cómo determinamos qué nodo o qué conexión ajustar?

Este es el problema de “asignamiento de crédito” (justamente el que nos conviene en nuestro particular problema). Back-propagation resuelve esto, al asumir que todos los elementos

de procesamiento y conexiones son de alguna forma culpables de una respuesta errónea.

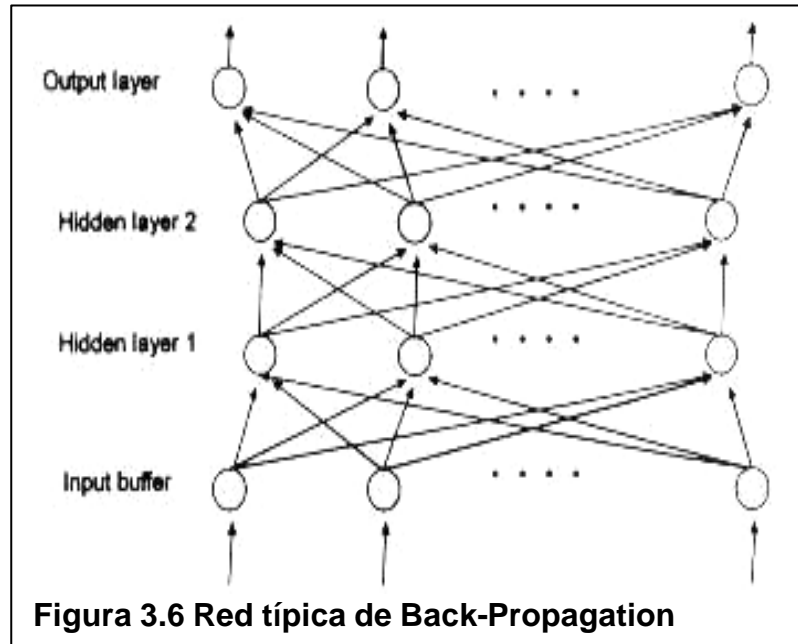
La responsabilidad del error del error se fija al propagar el error de salida hacia atrás a través de las conexiones de la capa previa. Este proceso se repite hasta que se llegue a la capa de entrada.

El nombre “back-propagation” se deriva de este método de distribuir la culpa por el error.

### **3.2.6.1 La red de Propagación hacia atrás**

La típica red de propagación hacia atrás siempre tiene una capa de entradas, una capa de salidas y por lo menos una capa oculta. No hay límite teórico para el número de capas ocultas pero típicamente serán una o dos. Algún trabajo se ha hecho que indica que un máximo de cuatro capas (tres ocultas y una de salida) son requeridas para resolver problemas

arbitrariamente complejos de clasificación de patrones. Cada capa está totalmente conectada con su p



Las flechas indican el flujo de la información durante la recordación. Durante el aprendizaje, la información es también propagada hacia atrás a través de la red y usada para actualizar los pesos de conexión. La red puede ser o hetero-asociativa o auto-asociativa, como se mencionó anteriormente.

Para evita el confundirnos de una capa a otra, una notación clara se hace necesaria para describir la regla de aprendizaje. Usamos un supraíndice en paréntesis rectos para indicar qué capa de la red está siendo considerada. El resto de la notación es como sigue:

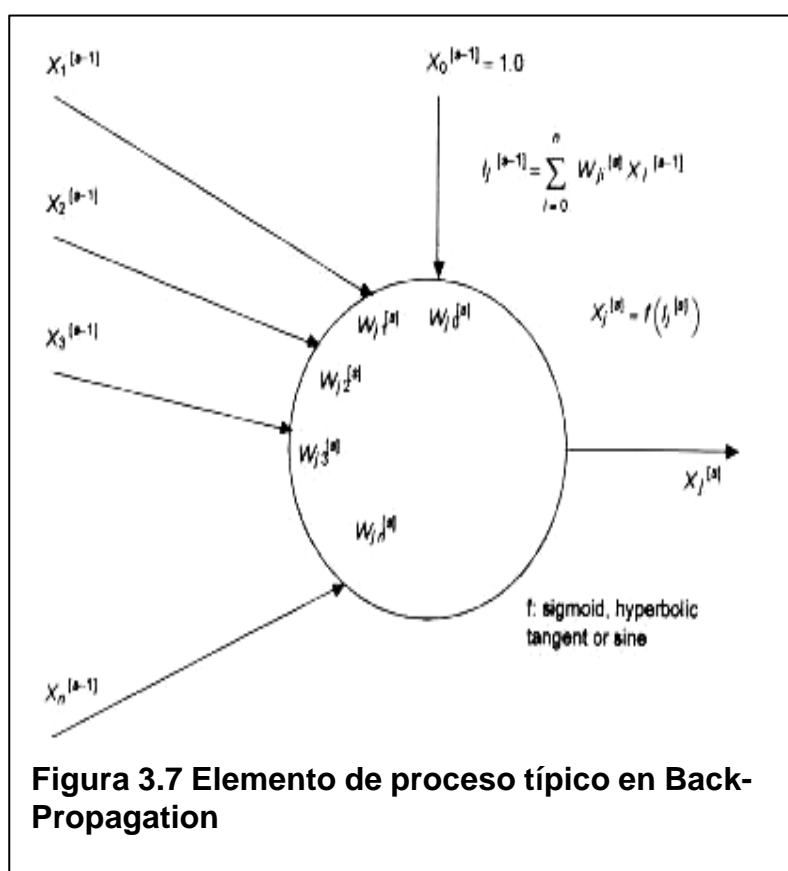
- $x_j^{[s]}$  actual estado de salida de la j-ésima neurona en la capa s.
- $w_{ji}^{[s]}$  peso en la conexión entre la i-ésima neurona de la capa (s-1) y la j-ésima neurona en la capa s.
- $I_j^{[s]}$  suma ponderada de las entradas hacia la j-ésima neurona en la capa s.

Un elemento de la red de propagación hacia atrás transfiere entonces sus entradas como sigue:

$$\begin{aligned} x_j^{[s]} &= f\left(\sum_i (w_{ji}^{[s]} \cdot x_i^{[s-1]})\right) \\ &= f(I_j^{[s]}) \end{aligned} \quad [1]$$

Donde  $f$  es tradicionalmente la función sigmoide pero puede ser cualquier función diferenciable. La función sigmoide se define como

$$f(z) = (1.0 + e^{-z})^{-1} \quad [2]$$



**Figura 3.7 Elemento de proceso típico en Back-Propagation**



### 3.2.6.2 Propagando hacia atrás el error local

Supongamos ahora que la red tiene alguna función de error global  $E$  asociada a ella, que sea una función diferenciable de todas las ponderaciones en la red. La función de error actual no es importante para entender el mecanismo de propagación inversa. El parámetro crítico que se pasa hacia atrás a través de las capas se define como

$$e_j^{[s]} = -\partial E / \partial I_j^{[s]} \quad [3]$$

Veremos después que esto se puede considerar como una medida del error local en el  $j$ -ésimo nodo en el nivel  $s$ .

Utilizando la regla de la cadena dos veces seguidas nos da una relación entre el error local y un nodo en particular en el nivel  $s$  y los errores locales en los niveles  $s+1$ :

$$e_j^{[s]} = f'(I_j^{[s]}) \cdot \sum_k (e_k^{[s+1]} \cdot w_{kj}^{[s+1]}) \quad [4]$$

Nótese que en la expresión [4], hay una capa sobre la capa  $s$ ; por tanto, [4] sólo puede ser utilizada para capas internas.

Si la función sigmoide se define en [2], entonces su derivada puede ser expresada como una función simple de sí misma como sigue

$$f'(z) = f(z) \cdot (1.0 - f(z)) \quad [5]$$

Por lo tanto, de [1], la ecuación [4] puede ser reescrita como

$$e_j^{[s]} = x_j^{[s]} \cdot (1.0 - x_j^{[s]}) \cdot \sum_k (e_k^{[s+1]} \cdot w_{kj}^{[s+1]}) \quad [6]$$

Dado que la función de transferencia es una sigmoide. El término de la sumatoria en [6] que es utilizado para retro-propagar errores es análogo al término de suma en [1] que es utilizado para propagar hacia delante las

entradas por la red. Entonces el mayor mecanismo en una red back-propagation es el propagar las entradas por las capas hasta la capa de salida, determinar el error en la salida, y luego propagar los errores hacia atrás desde la capa de salida hasta la capa de entrada usando [6] o más generalmente [4]. La multiplicación del error por la derivada de la función de transferencia escala el error.

### 3.2.6.3 Minimizando el error local

El objetivo del proceso de aprendizaje es el minimizar el error global  $E$  del sistema al modificar los pesos. Esta subsección mostrará cómo hacer esto basándonos en el conocimiento del error local en cada nodo.

Dado el actual conjunto de ponderaciones  $w_i^{[s]}$ , necesitamos determinar cómo incrementar o decrementarlas de manera que decrezca el error

global. Esto puede hacerse usando una regla de descenso por el gradiente como sigue:

$$\nabla w_{ji}^{[s]} = -lcoef \cdot \left( \partial E / \partial w_{ji}^{[s]} \right) \quad [7]$$

Donde *lcoef* es un coeficiente de aprendizaje. En otras palabras, cambiar cada peso de acuerdo al tamaño y dirección de el gradiente negativo en la superficie de error.

Las derivadas parciales en [7] pueden calcularse directamente del valor del error local discutido en la anterior subsección, porque, por la regla de la cadena y [1]:

$$\begin{aligned} \partial E / \partial w_{ji}^{[s]} &= \left( \partial E / \partial I_j^{[s]} \right) \cdot \left( \partial I_j^{[s]} / \partial w_{ji}^{[s]} \right) \\ &= -e_j^{[s]} \cdot x_i^{[s-1]} \end{aligned} \quad [8]$$

Combinando [7] y [8] tenemos

$$\nabla w_{ji}^{[s]} = lcoef \cdot e_j^{[s]} \cdot x_i^{[s-1]} \quad [9]$$

#### 3.2.6.4 La función de error global

La discusión previa a asumido la existencia de una función de error global sin especificarla. Esta función es necesaria para definir los errores locales en la capa de salida para que estos puedan ser retro-propagados hacia el interior de la red.

Supongamos que un vector es presentado en la capa de entrada de la red y supongamos que la salida deseada  $d$  es especificada por un instructor. Sea  $o$  quien denote la salida producida por la red con su actual conjunto de pesos. Entonces una medida del error al lograr esa salida deseada está dada por

$$E = 0.5 \cdot \sum_k (d_k - o_k)^2 \quad [10]$$

Donde el subíndice  $k$  indexa los componentes de  $d$  y  $o$ . Aquí, el error local primario está dado por  $d_k - o_k$ . De la expresión [3], el “error local escalado” de cada nodo de la capa de salida está dado por

$$\begin{aligned}
 e_k^{(o)} &= -\partial E / \partial I_k^{(o)} \\
 &= -\partial E / \partial o_k \cdot \partial o_k / \partial I_k \\
 &= (d_k - o_k) \cdot f'(I_k)
 \end{aligned}
 \tag{11}$$

$E$ , como se definió en [10], define el error global de la red para un vector particular  $(i, d)$ . Una función en conjunto del error global puede definirse como la suma de todas las funciones de patrones específicos del error.

Cada vez que un particular vector  $(i, d)$  es mostrado, el algoritmo de back-propagation modifica los pesos para reducir ese componente en particular de la función de conjunto global de error.

### 3.2.6.5 Resumen del algoritmo estándar de propagación inversa

Dado un vector de entrada  $i$  y una salida deseada  $d$ , hacer lo siguiente:

1. Presente  $i$  a la capa de entrada de la red y propáguelo hacia la capa de salida para obtener un vector de salida  $o$ .
2. Mientras esta información está siendo propagada por la red, también preparará todas las entradas sumadas  $I_j$  y estados de salida  $x_j$  para cada nodo en la red.
3. Para cada nodo en la capa de salida, calcular el error local escalado como en [11] y luego calcular el peso delta usando [9].
4. Para cada capa  $s$ , comenzando por la capa previa a la de salida y terminando por la posterior a la de entrada, y para cada nodo en la capa  $s$ , calcular el

error local escalado como en [6] y luego calcule los pesos delta usando [9].

5. Actualice todos los pesos en la red al sumar el término delta a sus correspondientes pesos previos. A esta regla se le denomina Regla Delta Generalizada que es la ley de aprendizaje de una red de propagación hacia atrás.

### **3.2.6.6 Variaciones al algoritmo estándar**

#### **Término de momento**

Uno de los problemas de un algoritmo de descenso por el gradiente es el asignar una tasa de aprendizaje apropiada. El cambiar los pesos como una función lineal de las derivadas parciales como está definido en [7] hace la suposición que la superficie de error es localmente lineal, donde “localmente” se define por el tamaño del coeficiente de aprendizaje.



En puntos de alta curvatura, esta asunción de linealidad no cuadra y un comportamiento divergente puede ocurrir cerca de dichos puntos. Es entonces importante el mantener el coeficiente de aprendizaje pequeño y eludir tal comportamiento.

Por otra parte un coeficiente de aprendizaje pequeño puede conllevar hacia una aprendizaje lento. El concepto de “término de momento” fue introducido para resolver esta dicotomía. La ecuación de peso delta [9] se modifica para que la una porción del anterior peso delta sea alimentado a través del actual peso delta:

$$\nabla w_{ji}^{[s]}(t) = lcoef \cdot e_j^{[s]} \cdot x_i^{[s-1]} + momentum \nabla w_{ji}^{[s]}(t-1) \quad [12]$$

Esto actúa como un filtro de baja permisión (low-pass filter) en los términos de pesos delta dado que las tendencias generales se refuerzan mientras que el comportamiento oscilatoria se auto cancela. Esto

permite entonces un pequeño valor para el coeficiente de aprendizaje y un aprendizaje más rápido.

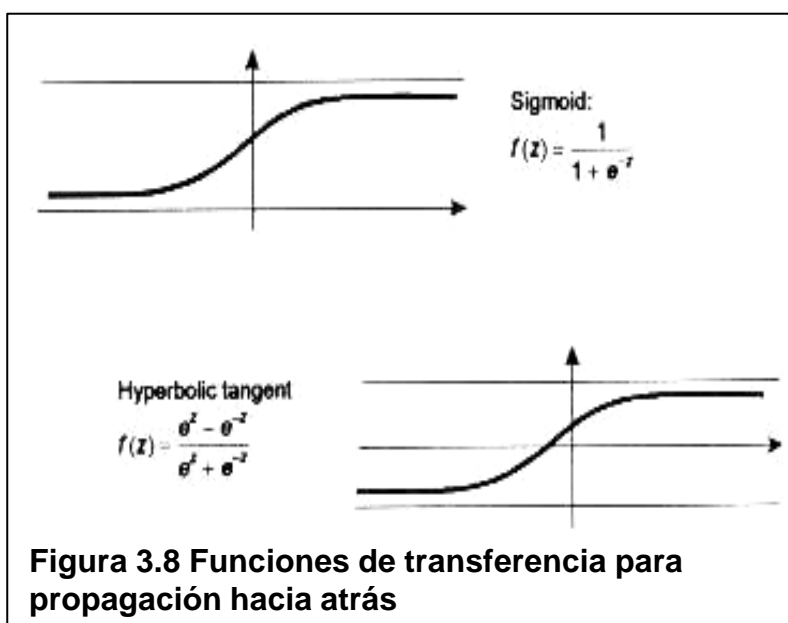
### **Actualización acumulada de los pesos**

Otra técnica que tiene un efecto en la rapidez de la convergencia es el solamente actualizar los pesos después de que los pares de (entrada, salida deseada) sean presentados a la red, en lugar que después de cada presentación. Esto es referido como propagación hacia atrás acumulada (cumulative back-propagation) porque los pesos delta son acumulados hasta que el conjunto completo de pares es presentado.

El número de pares de entrada / salida presentados durante la acumulación es denominada "época" (epoch). Esto puede corresponder sea al conjunto completo o a un subconjunto de pares de entrenamiento.

### Diferentes funciones de transferencia

En lugar de la función sigmoide, cualquier función “suave” puede ser utilizada como función de transferencia para un elemento de procesamiento. Comúnmente se cambia esta función por la de tangente hiperbólico, seno o lineal. Recuérdese que en un principio un perceptrón usaba una función umbral generalmente de tipo (0, 1). La tangente hiperbólica es solamente una versión bipolar de la sigmoide: la sigmoide (o logística) es una versión suave de la función umbral (0, 1) tanto que la tangente hiperbólica es una versión suave de una función umbral (-1, 1). Véase la figura 3.8.



La tangente hiperbólica se define por

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad [13]$$

Como en la función sigmoide, la derivada de la tangente hiperbólica puede también ser expresada en términos de si misma:

$$f'(z) = (1.0 + f(z)) \cdot (1.0 - f(z)) \quad [14]$$

Por tanto con este tipo de función de transferencia la función propagadora del error [4] se modifica en

$$e_j^{[s]} = (1.0 + x_j^{[s]}) \cdot (1.0 - x_j^{[s]}) \cdot \sum_k (e_k^{[s+1]} \cdot w_{kj}^{[s+1]}) \quad [15]$$

Esto como para ejemplificar la diversidad de los resultados al cambiar las funciones de transferencia. Gran parte de los paquetes neurocomputacionales actuales sugieren las funciones de transferencia por “default”. Sólo la experiencia al experimentar y la familiaridad con el problema harán modificar las funciones de transferencia, según ciertos expertos.

### 3.3 Observaciones

Todo lo antes expuesto tiene un solo fin: sustentar teóricamente la herramienta con que solucionaremos el problema de análisis crediticio. Una buena parte de la teoría existente busca solucionar problemas matemáticos de aproximaciones funcionales no lineales tales como el de la “O-Exclusiva”, un claro caso de análisis discriminante.

Las redes de “propagación hacia atrás” son necesariamente la opción más adecuada para ser utilizada en la determinación de si un cliente es o no es un buen sujeto de crédito. Esto ocurre primariamente por la capacidad de la red de imitar una correspondencia funcional entre un conjunto de entrada y otro de salida.

Tratemos de hacer una abstracción del problema: a la operadora celular Bellsouth acuden cierto número de clientes con diferentes características crediticias que podemos agrupar como variables en un vector  $i$  de entrada. Así mismo, luego de aceptar al cliente, este sujeto de crédito genera un historial en una base de datos de cobranzas.

Evaluando al cliente en función de su historial mas no de sus características crediticias podremos decir que tan “buen pagador” es el cliente a través de sus pagos registrados en la base de datos. Sean estos resultados agrupados en un vector  $d$  de salida.

Entonces tendríamos ya material suficiente para entrenar una red de back-propagation a reconocer los patrones de comportamiento de los clientes utilizando sus variables crediticias. Esta es la red que, una vez entrenada, podrá repetir dicho comportamiento y evaluará a un cliente con los datos de la hoja de riesgo de la que se hace mención en el capítulo 1.

## **CAPITULO IV**

### **4. ANÁLISIS Y APLICACIÓN DE UN MODELO DE REDES NEURONALES UTILIZANDO UN SOFTWARE PARA RESOLVER EL PROBLEMA DE CARTERA VENCIDA.**

#### **4.1 Introducción**

En este último capítulo procederemos a plantear en específico y resolver el problema de neuroclasificación de los datos obtenidos de Bellsouth al someterlos a un proceso de aprendizaje de un modelo de redes neuronales previamente diseñado y por medio de un paquete computacional obtendremos un resultado esperando que este sea positivo. Así, al final de este capítulo podremos ya tener un proceso neuro-discriminatorio artificial para diferenciar posibles sujetos de crédito de los que probablemente no lo sean.

## 4.2 Sobre los datos

Si observamos la Hoja de Administración de Riesgo del capítulo 1, podremos saber que obviamente la información que vamos a utilizar proviene única y exclusivamente de ahí. Hemos tomado en cuenta sólo 6 variables cuyo único resultado puede ser: “El cliente aparece eventualmente en mora o no”.

Las variables que utilizamos son:  $i_1$ : Edad (entero),  $i_2$ : Estado Civil (boolean),  $i_3$ : Casa Propia / alquilada (boolean),  $i_4$ : Trabajo Empleado-Operativo / Ejecutivo-Propietario (boolean),  $i_5$ : Años en el último trabajo (entero),  $i_6$ : Pago con Débito-Cuenta / Débito-Tarjeta (boolean). La variable de resultado es  $d_1$ : Aparece eventualmente en mora por más de un mes (boolean).

Estos datos constituyen una muestra de aquellos clientes cuyos planes tarifarios son cuentas individuales de Bellsouth guayaquil del período julio 2000 – julio 2001. El total de la muestra es  $n = 100$  sujetos. No se toma en cuenta ningún caso de planes de prepago por obvias razones.



La muestra es tan reducida debido a lo confidencial de los datos y sobretodo a la mala calidad de la información. En todos los casos no había mas que papeles (cero registros digitalizados) y en la mayoría había datos incompletos o ilegibles; también debemos tener en cuenta que es muy probable que exista algún tipo de sesgo.

Evidente es que no tomamos en cuenta la variable “ingresos” (que hubiera sido de suprema importancia) ya que nunca este campo fue llenado para caso alguno. El resto de variables fueron descartadas por su poca influencia dentro del modelo (i.e. nombre, apellido, teléfono, dirección, nombre del cónyuge, etc.).

Aquellos registros donde aparecen ceros en el campo “deuda” significan que de lo que lleva el cliente al cabo de ese año, este no aparece con deudas por mora (más allá del mes) dentro de los registros de Bellsouth.

Cabe mencionar que esta es la única información digitalizada que se posee. En esta parte hay que tomar en cuenta lo complicado que se torna el obtener esta información por lo delicado de su contenido.

Hemos dividido a la mitad el conjunto de observaciones de tal forma que el primer subconjunto nos sirva para entrenar la red y la otra mitad para efectos de prueba. En los anexos podemos encontrar esta información.

La información originalmente fue digitada en MSExcel pero los subconjuntos necesariamente fueron importados a dos archivos de texto distintos (\*.txt) para poder utilizarlos en el programa que escogimos.

### **4.3 Acerca del Software utilizado**

Después de una ardua búsqueda por algún “Trial version” software de redes neuronales (ya que casi la totalidad existente requiere de licencia y esta es muy costosa) optamos por emplear el programa

***ThinksPro–Neural Networks for Windows Trial Version 1.05*** de  
***Logical Designs Consulting.***

Este programa de 1995, por antiguo que parezca presta todos los requerimientos necesarios para llevar a cabo nuestro trabajo. Ofrece las facilidades de crear nuevos proyectos, crear redes, editarlas, parametrizarlas, medir el error al iterar, ofrecer y exportar los resultados, etc.

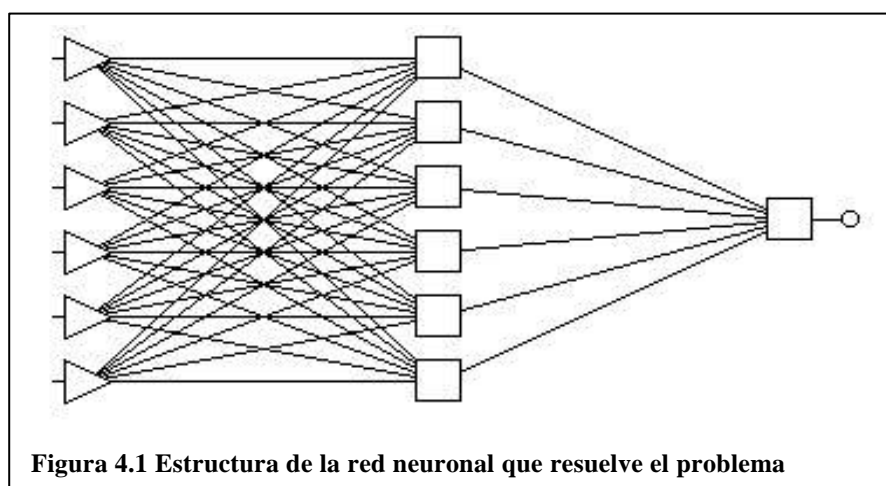
Vamos nosotros a mencionar paso a paso el procedimiento que seguimos para ejecutar el entrenamiento de aquella red neuronal que creamos para hacer el reconocimiento de los patrones crediticios.

#### **4.4 Proceso de Aplicación del modelo sobre los datos**

Los conjuntos de datos tienen por denominación prueba11.txt y prueba22.txt y se pueden obtener de los anexos.

#### 4.4.1 Creación de la red

Para crear una red neuronal nueva primero creamos un “proyecto nuevo” tal y como crearíamos un documento nuevo en un procesador de palabras. La primera tarea a cumplir será diseñar y parametrizar una estructura neuro-artificial que concuerde con la lógica de nuestro problema y que utilice la regla de aprendizaje que hemos estudiado.



Como podemos observar en la figura 4.1, esa es la estructura o “esqueleto” de la red que utilizaremos: 3 capas, una de entrada con 6 nodos, una intermedia también con 6 nodos y una de salida con un único nodo. Todos los nodos de la capa

anterior se interconectan con todos los nodos de la capa posterior, como es usual en Back-Propagation.

La simple lógica nos explica por qué tenemos 6 nodos en la capa de entrada y 1 en la capa de salida: cada nodo de la capa de entrada representará la entrada de cada variable de la hoja de riesgo, esto es  $i_1, i_2, \dots, i_6$ ; y el nodo de la capa de salida representa el resultado del análisis crediticio  $d_1$ . El número de capas ocultas y nodos en la misma provienen de razones un tanto distintas.

Por experiencia de anteriores trabajos de distintos autores y distintos archivos de ayuda de muchos programas, podemos decir que en general, si el “tamaño del problema” es pequeño entonces no es necesario sobredimensionar la red con más nodos ni capas ocultas de lo proporcional al resto de la red.

En todo caso consideramos que el número de capas y nodos en la sección intermedia de la red que hemos utilizado en este

problema han sido el máximo posible basados en la experiencia de trabajos de distintos autores.

Después de determinar la estructura procedemos a escoger la regla de aprendizaje de la red y sus parámetros. De la hoja de "Network Setup" escogemos "back-propagation" para nuestra regla de aprendizaje.

Como la capa de entrada sirve únicamente como "input buffer" entonces la única parametrización en esa capa es el preprocesamiento de entrada como MAX/MIN, esto sirve para transformar los datos hacia un intervalo decimal fijo considerando los valores máximos y mínimos de cada variable y recodificándolos al hacerlos variar dentro de esos intervalos. Esto se hace debido a que presta facilidad a los cálculos y se acomoda mejor a la función de transferencia que es umbraloides en la capa oculta.

La figura 4.2 muestra el diseño de la capa de entrada. En la capa oculta procedemos a parametrizarla conforme lo antes establecido. Tenemos 6 nodos, utilizamos una suma ponderada (en la figura se ve: dot product) y una función de transferencia sigmoide (establece una especie de umbral 0-1).

The screenshot shows the 'Neural Network Setup' dialog box with the following configuration:

- Architecture:** Multilayer Normal Feed Forward
- Error Type:** Mean Square Error
- Hidden Layers:** 1
- Random Seed:** 777
- Batch Size:** 1 (selected), +50, Training Set
- Layer Selection:**
  - Input Layer (MaxMin)
  - Hidden Layer 1 (BPN)
  - Hidden Layer 2
  - Hidden Layer 3
  - Output Layer (BPN)
- Layer Definition:**
  - Inputs:** 6
- Input Preprocessing:** Max Min Processing (MaxMin)
- Sample Arrangement:**
  - Normal
  - Time Series: 2 consecutive samples
- Global Parameters:**
  - Input Noise Factor (InputNoise):** 0
  - Training Error Tolerance (ErrorTol):** 0

Buttons: OK, Cancel, Defaults...

Figura 4.2 Parametrización en la capa de entrada

En la figura 4.3 vemos además que las tasas de velocidad de aprendizaje y de momento antes discutidas se han elegido con valores de 0.09 y 0.07 respectivamente.

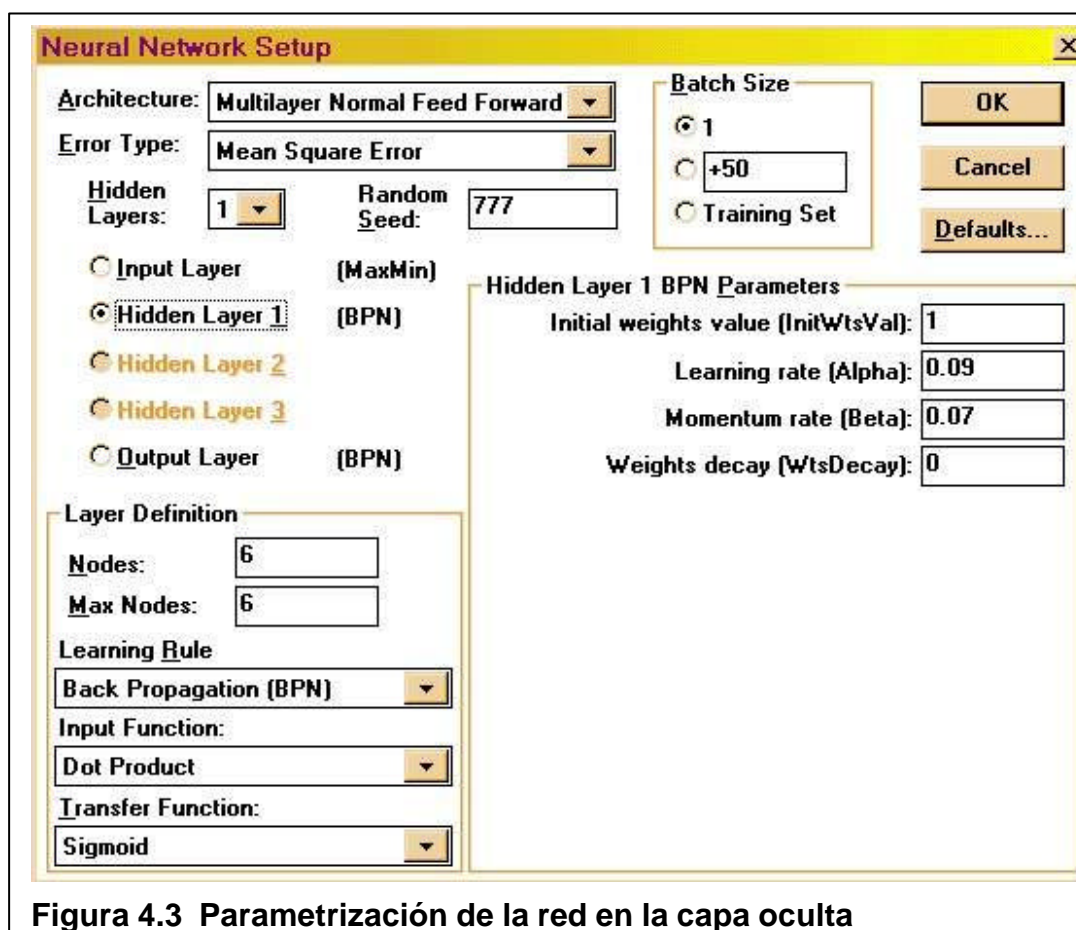


Figura 4.3 Parametrización de la red en la capa oculta

Estos valores fueron elegidos conforme hicimos algunos ensayos hasta encontrarlos como los más adecuados al equilibrar la velocidad del aprendizaje y el riesgo de caer en mínimos locales.



En el primer intento por elegir una tasa de aprendizaje optamos por 0.01 y probamos que era una tasa muy baja ya que el aprendizaje era muy lento.

La mayor parte del aprendizaje la hace la capa oculta en este caso debido al mayor número de nodos que esta tiene con respecto a la capa de salida.

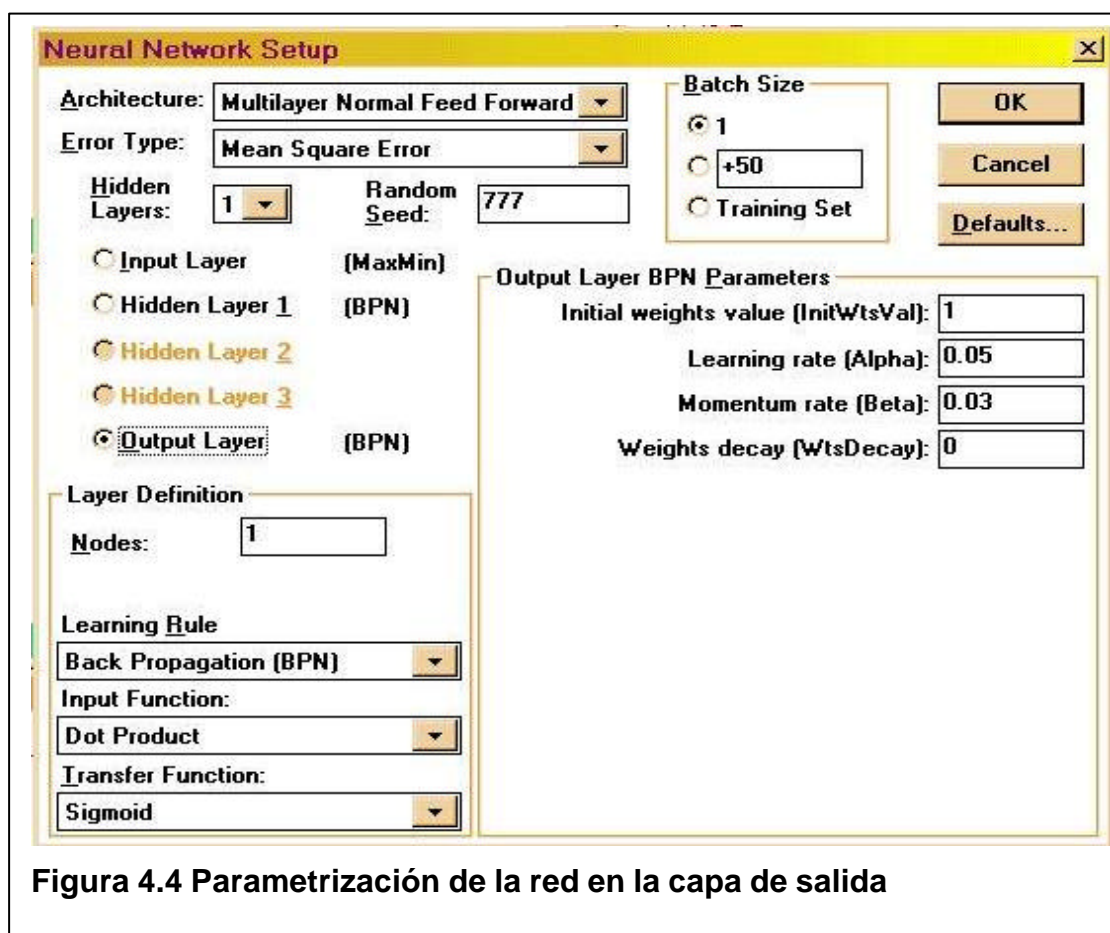


Figura 4.4 Parametrización de la red en la capa de salida

En la figura 4.4 observamos la parametrización de la capa de salida. Se ve un único nodo cuya función de resultado o transferencia también es una sigmoide. Nótese que en cada capa se puede elegir la regla y parámetros de aprendizaje independientemente de las otras capas, esto es de cierto modo una característica única del programa que hemos utilizado ya que la mayoría no permite tanta interactividad con el diseño de las redes.

Esto se debe a que los creadores de los programas los hacen con el fin de quien tiene muy pocas nociones sobre el diseño de las redes neuronales puedan acceder a esta tecnología por medio de “asistentes virtuales” como los tiene por ejemplo PowerPoint al crear una presentación.

Esto obviamente es desventajoso hasta cierto punto para quien desea total interactividad con el diseño debido a su familiaridad con el mismo. La ventaja es el ahorro de tiempo.

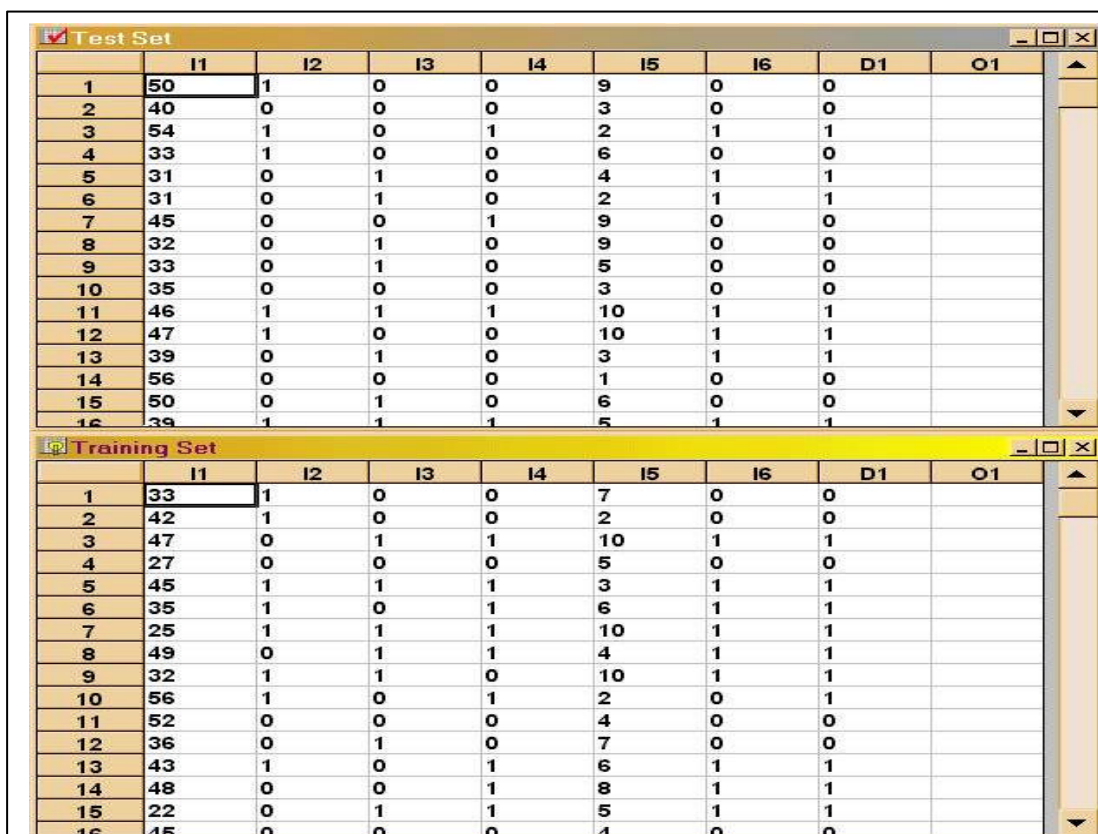
De la misma figura 4.4 nótese que los parámetros de velocidad y momento escogidos fueron 0.05 y 0.03 respectivamente. La razón de ser de estos valores es similar a la expuesta en la capa oculta con la diferencia que en esta capa la velocidad del aprendizaje no es muy influenciada debido a la existencia de un único nodo.

#### 4.4.2 Importación de los datos

El procedimiento de importación de datos de entrenamiento y prueba es bastante sencillo. Lo que hay que cuidar es digitar la información previamente de tal manera que ThinkPro pueda reconocerla sin ningún problema. Por ejemplo no se debe grabar los datos con el encabezado como nombre de la variable ya que el programa lo hace por omisión.

Simplemente optamos por el comando **File / Import File / Training set** y **File / Import File / Test set** y para importar los archivos prueba11.txt y prueba22.txt que previamente guardamos desde MSExcel.

Después de cada importación la hoja de datos que al principio estuvo vacía se llena automáticamente con los datos digitados, también las variables reciben su nombre automáticamente y ThinksPro considera de manera automática que la última variable es la de salida, por eso es necesario que conservemos ese orden al ingresar los datos.



The image shows two data sheets from the ThinksPro software. The top sheet is titled 'Test Set' and the bottom sheet is titled 'Training Set'. Both sheets have columns labeled I1, I2, I3, I4, I5, I6, D1, and O1. The 'Test Set' sheet contains 16 rows of data, and the 'Training Set' sheet contains 16 rows of data.

|    | I1 | I2 | I3 | I4 | I5 | I6 | D1 | O1 |
|----|----|----|----|----|----|----|----|----|
| 1  | 50 | 1  | 0  | 0  | 9  | 0  | 0  |    |
| 2  | 40 | 0  | 0  | 0  | 3  | 0  | 0  |    |
| 3  | 54 | 1  | 0  | 1  | 2  | 1  | 1  |    |
| 4  | 33 | 1  | 0  | 0  | 6  | 0  | 0  |    |
| 5  | 31 | 0  | 1  | 0  | 4  | 1  | 1  |    |
| 6  | 31 | 0  | 1  | 0  | 2  | 1  | 1  |    |
| 7  | 45 | 0  | 0  | 1  | 9  | 0  | 0  |    |
| 8  | 32 | 0  | 1  | 0  | 9  | 0  | 0  |    |
| 9  | 33 | 0  | 1  | 0  | 5  | 0  | 0  |    |
| 10 | 35 | 0  | 0  | 0  | 3  | 0  | 0  |    |
| 11 | 46 | 1  | 1  | 1  | 10 | 1  | 1  |    |
| 12 | 47 | 1  | 0  | 0  | 10 | 1  | 1  |    |
| 13 | 39 | 0  | 1  | 0  | 3  | 1  | 1  |    |
| 14 | 56 | 0  | 0  | 0  | 1  | 0  | 0  |    |
| 15 | 50 | 0  | 1  | 0  | 6  | 0  | 0  |    |
| 16 | 39 | 1  | 1  | 1  | 5  | 1  | 1  |    |

|    | I1 | I2 | I3 | I4 | I5 | I6 | D1 | O1 |
|----|----|----|----|----|----|----|----|----|
| 1  | 33 | 1  | 0  | 0  | 7  | 0  | 0  |    |
| 2  | 42 | 1  | 0  | 0  | 2  | 0  | 0  |    |
| 3  | 47 | 0  | 1  | 1  | 10 | 1  | 1  |    |
| 4  | 27 | 0  | 0  | 0  | 5  | 0  | 0  |    |
| 5  | 45 | 1  | 1  | 1  | 3  | 1  | 1  |    |
| 6  | 35 | 1  | 0  | 1  | 6  | 1  | 1  |    |
| 7  | 25 | 1  | 1  | 1  | 10 | 1  | 1  |    |
| 8  | 49 | 0  | 1  | 1  | 4  | 1  | 1  |    |
| 9  | 32 | 1  | 1  | 0  | 10 | 1  | 1  |    |
| 10 | 56 | 1  | 0  | 1  | 2  | 0  | 1  |    |
| 11 | 52 | 0  | 0  | 0  | 4  | 0  | 0  |    |
| 12 | 36 | 0  | 1  | 0  | 7  | 0  | 0  |    |
| 13 | 43 | 1  | 0  | 1  | 6  | 1  | 1  |    |
| 14 | 48 | 0  | 0  | 1  | 8  | 1  | 1  |    |
| 15 | 22 | 0  | 1  | 1  | 5  | 1  | 1  |    |
| 16 | 45 | 0  | 0  | 0  | 4  | 0  | 0  |    |

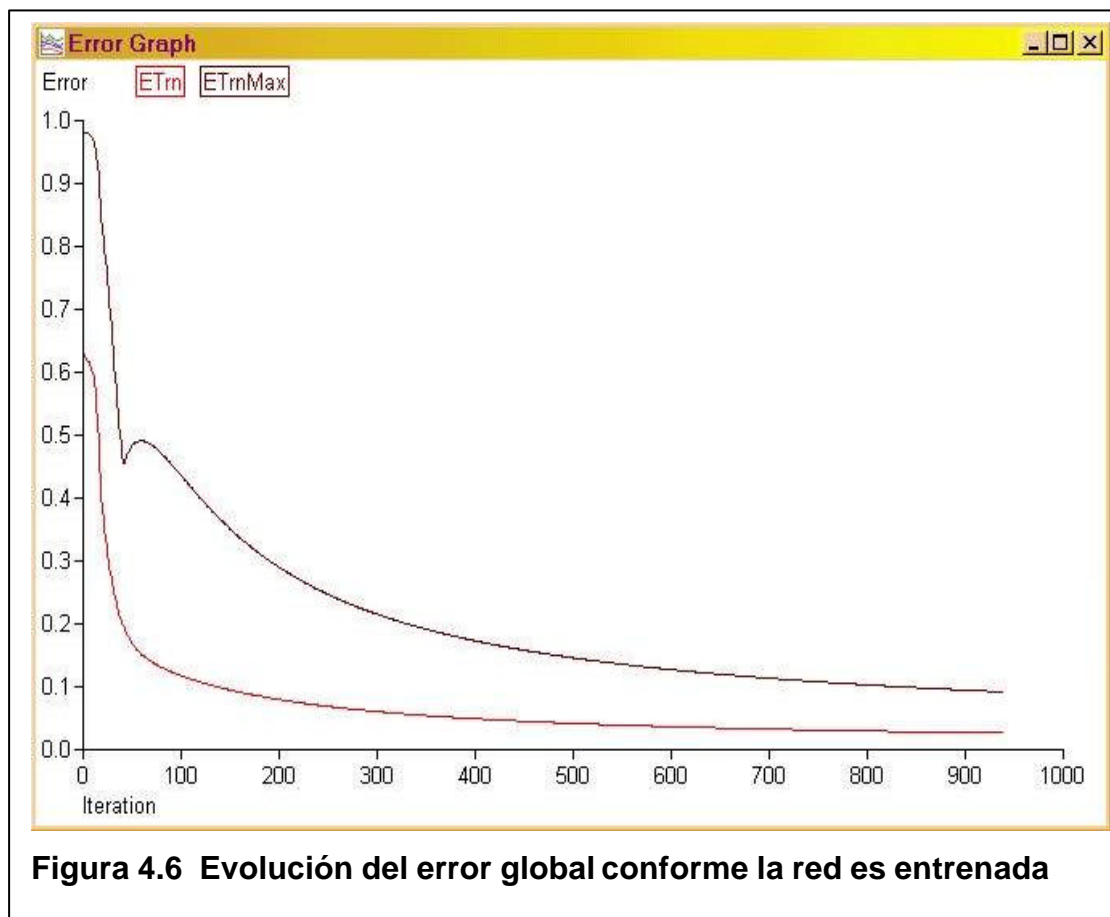
**Figura 4.5 Muestra de las hojas de datos en ThinksPro**

En la figura 4.5 observamos el caso para los conjuntos de entrenamiento y prueba.

La columna  $o_1$  representa el valor que la red le dará al resultado producto del entrenamiento, así, el error se construye a partir de la diferencia entre el valor deseado  $d_1$  y el valor actual  $o_1$ . El valor inicial de  $o_1$  está dado por una semilla aleatoria puesta en los valores de los pesos de los nodos en las distintas capas.

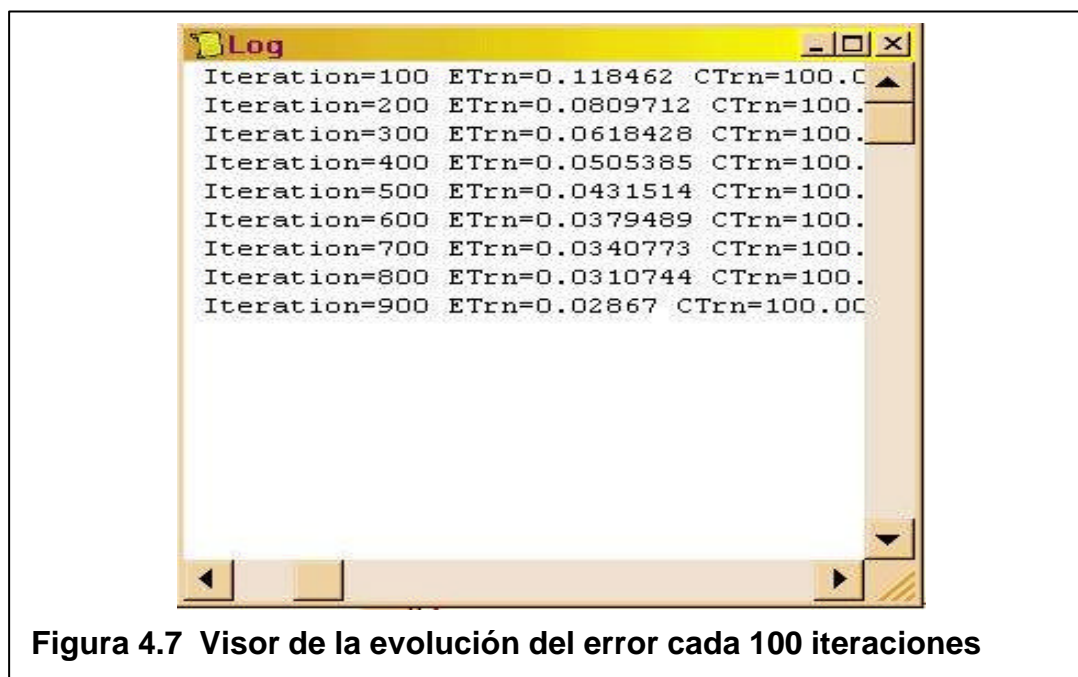
#### 4.4.3 Entrenamiento de la red

Inicialmente es posible probar la red o entrenarla (opción abierta que el programa permite) pero suena ilógico probar una red que no ha sido entrenada. Entonces procedemos a habilitar la opción de entrenamiento. Para esto hemos elegido un valor mínimo del error de 0.028 como criterio de convergencia. Elegimos también que el número de iteraciones sea infinito hasta que se encuentre bajo convergencia. Elegimos **Run** para el conjunto de entrenamiento y procedemos a entrenar la red.



En la figura 4.6 se puede ver que el error baja rápidamente hasta que encuentra el criterio de convergencia. La línea de color roja representa el error al que nos referimos: **ETrn** o “error de entrenamiento”.

La evolución del error también se captura en un archivo, la figura 4.7 muestra parte del visor de los valores de error de entrenamiento en ThinksPro.

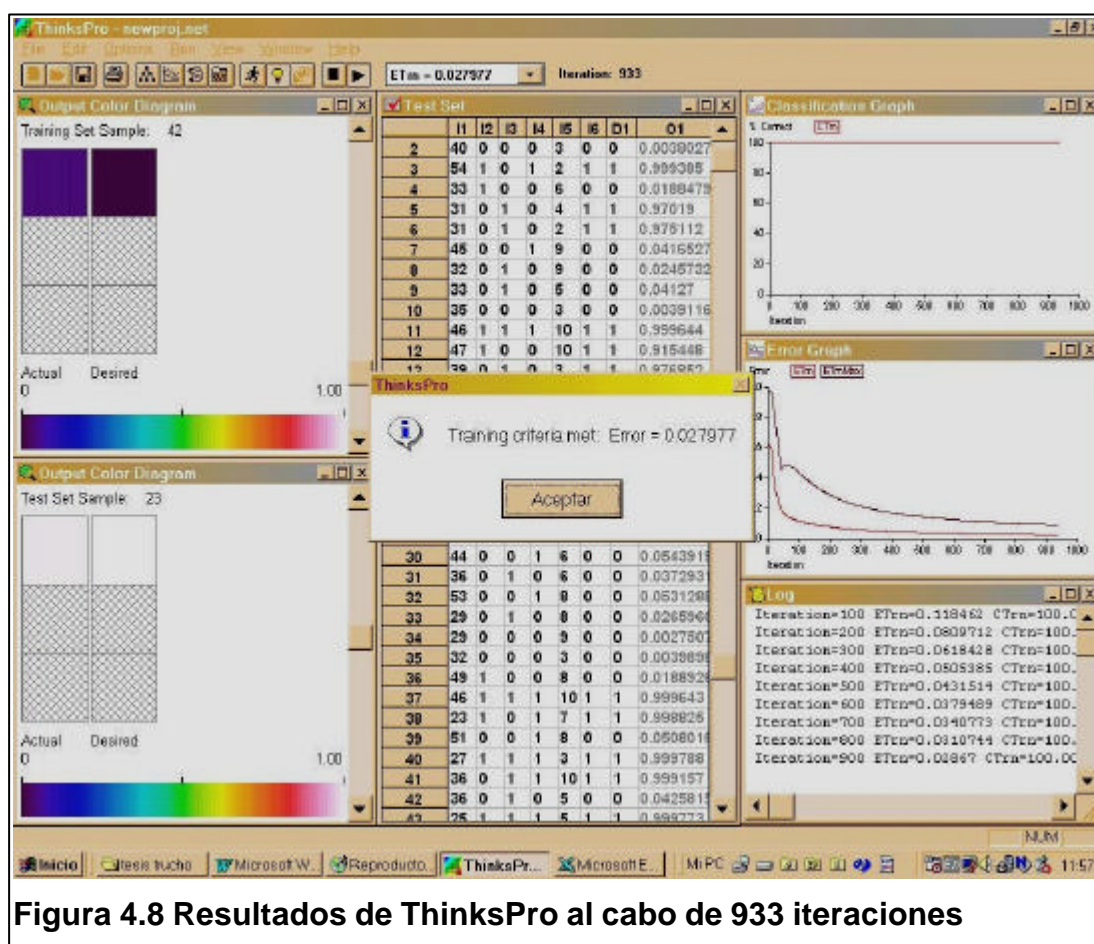


**Figura 4.7** Visor de la evolución del error cada 100 iteraciones

El valor final del error no se muestra en este visor ya que cuando se llega a convergencia el programa muestra un mensaje afirmándolo.

## 4.5 Obtención de los resultados

Parte de los resultados ya los hemos mostrado en lo que respecta del valor final del error. Observemos la figura 4.8.



**Figura 4.8 Resultados de ThinksPro al cabo de 933 iteraciones**

Notemos el mensaje que ThinksPro muestra al final. El error, al cabo de 933 iteraciones sobre el conjunto de datos de entrenamiento es de 0.027977 que ciertamente es inferior al error propuesto de 0.028.



Hay dos cosas más de interés que podemos observar en la figura 4.8: la primera es aquellos los valores que adquieren las variables de salida  $\mathbf{o}_1$  para los conjuntos de entrenamiento y prueba.

Para los casos en que el resultado deseado es “deudor” (es decir,  $\mathbf{d}_1 = 1$ ) el valor actual de  $\mathbf{o}_1$  es muy cercano a 1, y para los casos en que el resultado deseado es “no deudor” ( $\mathbf{d}_1 = 0$ ) el valor actual de  $\mathbf{o}_1$  es muy cercano a cero; cosa que constituye otra prueba que el entrenamiento ha sido eficaz.

La segunda cuestión de interés es el visor de resultados de ThinksPro. De manera novedosa se crea una escala de colores que varia desde cero hasta uno. Esta escala representa todos los posibles valores que  $\mathbf{o}_1$  puede tomar. Caso por caso se puede ver como aquellos valores cercanos a cero tienen un color muy oscuro (negro = 0) y que los valores cercanos a uno tienen un color claro (blanco = 1).

Este visor existe para ambos conjuntos, de entrenamiento y de prueba. En la figura 4.8 observamos los colores que tienen los valores de  $d_1$  y  $o_1$  para los casos 42 y 23 de entrenamiento y prueba respectivamente.

Cuando la red no está entrenada los colores no coinciden y rara vez se parecen, en este caso tenemos todo lo contrario.

Al final tenemos a éste como último criterio para establecer la eficacia del entrenamiento de la red. En los anexos podemos ver el archivo de resultados que concuerdan con lo aquí expuesto.

## **4.6 Observaciones**

### **4.6.1 Sobre el conjunto de datos**

Una observación final en este capítulo es que hemos detectado cierto sesgo. Lo bueno de esto es que si tenemos un nivel de certeza del origen del mismo.

El sesgo consiste en que en la mayoría de los casos aquellos clientes que pagan a través de su tarjeta de crédito se reportan como deudores en la muestra.

Esto obviamente distorsiona el aprendizaje ya que resulta ilógico el generalizar esta premisa. Con esto queremos decir que tal y como está entrenada la red no fuera suficiente (ni confiable) para que esta evalúe a un nuevo cliente y lo clasifique dentro del grupo deudor o no deudor.

El origen de este sesgo proviene de las pocas fuentes confiables de datos que obtuvimos. Lo lógico es pensar que la mayoría de deudores que digitamos provenían de algún archivo específicamente de tarjeta habientes.

Es el riesgo que asumimos al llevar a cabo este trabajo. En las conclusiones trataremos este tema con mayor amplitud.

#### 4.6.2 Sobre el uso práctico de la red

Otra observación será mas bien el explicar como sería la utilización práctica de la red. Supongamos que la red esta bien entrenada (como de hecho está), con un conjunto confiable de datos (cosa difícil en este caso), entonces supongamos también que somos el departamento de crédito de Bellsouth, encargado de aprobar y desaprobar los planes tarifarios otorgados a los nuevos clientes.

La red neuronal entrenada me servirá para que de ahora en adelante yo pueda tener un mejor criterio de a quién aprobarle los planes y a quién no, de la siguiente manera: En el conjunto de datos de Prueba inserto un nuevo caso.

Este nuevo registro representa al nuevo potencial cliente y sus variables crediticias a ser evaluadas. En el campo  $d_1$  no debemos ingresar valor alguno ya que no sabemos a priori si es que este nuevo sujeto será deudor o no.

Al ejecutar el archivo de prueba desde su visor de datos obtenemos inmediatamente un valor para  $\mathbf{o}_1$ . Al hacer la misma ejecución desde el visor de colores obtenemos lo mismo, que es un nuevo color (o valor) para  $\mathbf{o}_1$ .

Entonces si este nuevo valor se aproxima a 1 sabremos que probablemente este cliente no sea un buen pagador y si se aproximara a cero pensáramos lo contrario.

Es así como finalmente hemos utilizado una tecnología complicada en una idea sencilla. Obviamente que a medida que la base de datos es alimentada habría que reentrenar la red periódicamente.

## CONCLUSIONES Y RECOMENDACIONES

1. Primeramente, podemos afirmar que lo que acabamos de hacer no puede generalizarse para cualquier caso ya que dependiendo de que datos se tenga y su relación real con el resultado esperado se esperará un mejor o peor entrenamiento y por ende una respuesta distinta de la red. Es obvio que en el fondo esto debe ser un juego de correlaciones.

No podemos negar que aunque la red no emplea necesariamente métodos estadísticos conocidos para establecer el grado de cambio que deba dar a sus pesos no es menos cierto que ésta establece una relación intrínseca entre las variables de entrada y las de salida.

Así, lo que queremos decir es que entre las seis variables de entrada y la variable de salida existe de hecho una matriz de correlaciones y que el grado de aportación que cada una de éstas hace al cambio o

variación de la variable de salida es ahí medido, y que ya que esto es verdad también es verdad que podríamos medir el grado de influencia sobre la variación de la variable de salida en un nuestro modelo de redes neuronales para así saber que variables de entrada son las más “importantes” en relación a su influencia sobre los resultados.

El asunto es aquí es saber que podemos saber a priori si una variable nos va a ser útil o solo será un relleno en el modelo. Lo que ha ocurrido entonces en nuestro caso tal vez no sea producto de la “bondad” de todas las seis variables, sino que muy probablemente la variable que domine la relación con respecto del resultado sea aquella que nos está sesgando la muestra:  $i_6$  o “el cliente paga por débito de cuenta o tarjeta de crédito”.

Entonces podemos recomendar para futuros trabajos e investigaciones el hacer un análisis de correlaciones antes de proseguir con el modelamiento de la red, claro está, esto no es estrictamente necesario ya que la red aprenderá más de las variables que más aporten sin importar el resto de las variables. Además no

podemos pedir que todo modelador de redes y peor que todo analista de crédito sea estadístico.

2. En este trabajo, lo más importante en la práctica será recomendar que la toma de datos sea seria. A nuestro parecer las hojas de administración de riesgo se llenan como una simple formalidad sin darle importancia alguna a la información que ahí debe incluirse. Campos vacíos, llenado ilegible, información incierta y no comprobada no pueden ser las variables que nos lleven a un análisis crediticio coherente.

Recomendamos que se lleven registros digitalizados con la información tomada y que no se llenen papeles para abandonarlos en alguna bodega, por increíble que parezca esto es lo que ocurría y ya vemos cuál es el resultado de esto.

3. También es importante el afirmar que las variables que se usan en la hoja de administración de riesgo no son necesariamente las adecuadas. De hecho, según autores en análisis multivariante lo



recomendable en el caso crediticio será utilizar, tamaño de la familia, ingresos, avalúo de la casa, edad y años de trabajo. Vemos entonces que gran parte de este grupo de variables no es utilizado en Bellsouth.

4. En lo que respecta a la red mencionamos anteriormente que decidimos usar la regla de aprendizaje de Back-propagation para entrenar a la red. Existe sólo un justificativo matemático para utilizar esta regla: la capacidad de aproximar una relación funcional.

Como por hipótesis las variables de entrada están relacionadas no linealmente con la variable de salida entonces tenemos aquí la razón por la que la propagación inversa da una respuesta al análisis.

5. Aunque encontramos deficiencias en los resultados debido al sesgo hemos encontrado que este proceso de análisis es casi ya una necesidad para un análisis crediticio, esta conclusión no se basa en los resultados necesariamente, se basa en la coherencia que el proceso de aprendizaje tiene. La red neuronal realmente aprende cual si fuera un ente inteligente, esa es la idea de la "inteligencia" artificial.