

# Comparación de Modelos Neuronales Utilizadas en Sistemas de Soporte para la Toma de Decisiones

Carlos Manuel Vicentín<sup>1</sup>, Darío Abelardo Quintana<sup>1</sup>, y Juan Carlos Insfrán<sup>1</sup>

<sup>1</sup> Alumnos de la cátedra Inteligencia Artificial de Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Resistencia

[carlosvicentin@yahoo.com.ar](mailto:carlosvicentin@yahoo.com.ar)

[darioquintana@gmail.com](mailto:darioquintana@gmail.com)

[juancainsfran@yahoo.com.ar](mailto:juancainsfran@yahoo.com.ar)

**Resumen.** El presente trabajo consiste en la comparación de dos tipos de redes neuronales, Backpropagation y Hopfield, ante un escenario planteado por la cátedra Inteligencia Artificial de la Facultad Regional Resistencia UTN. Luego del correspondiente marco teórico de introducción a las redes neuronales se presentará la propuesta de implantación con sus respectivos problemas en donde la finalidad de ambas redes será reconocer según un modelo de cliente determinado si una persona que solicita un crédito se corresponde con unos de los posibles perfiles predefinidos (Apto, No Apto y Dudoso en el caso que se requiera evaluar algún aspecto subjetivo del cliente). Por ultimo en la conclusión se reflejará mediante tablas y gráficos estadísticos el funcionamiento de las dos redes.

## 1 Introducción

Con las Redes Neuronales se busca la solución de problemas complejos, no como una secuencia de pasos, sino como la evolución de unos sistemas de computación inspirados en el cerebro humano, y dotados por tanto de cierta "inteligencia", los cuales no son sino la combinación de elementos simples de proceso interconectados, que operando de forma paralela en varios estilos, consiguen resolver problemas relacionados principalmente con la clasificación y reconocimiento de patrones. Este tipo de solución es la más adecuada para la implementación de sistemas de soporte para la toma de decisión, tal como el tipo de escenario que se plantea en el presente informe.

## 2 Marco Teórico

A continuación se presenta una breve introducción teórica respecto a los tipos de redes neuronales con las cuales se dio solución al escenario presentado por la cátedra.

## 2.1 Red de Hopfield

Las redes Hopfield son redes monocapa con  $N$  neuronas. Sus valores de salida son binarios: 0/1 ó -1/1, para el modelo original llamado DH (Discret Hopfield), siendo las funciones de activación de las neuronas del tipo escalón. Esto quiere decir que si la suma de las entradas de las neuronas es mayor o igual que el umbral, la activación es 1. Si es menor, la activación es 0 (o -1) [Hilera, J. R., Martínez, 2000].

Sus capacidades están limitadas, debido a que la función escalón no puede definir la derivada en el punto de transición y esto no ayuda a los métodos de aprendizaje en los cuales se usan derivadas [Ana Bollella, 1997].

Otra característica de este tipo de redes es que no son auto-recurrentes, ya que las salidas de las neuronas se comunican con todas las demás pero no consigo misma [Hilera, J. R., Martínez, 2000].

### 2.1.1 Aprendizaje

Las redes Hopfield utilizan aprendizaje OFF-LINE, esto quiere decir que el aprendizaje de la red supone la desconexión de la misma. De esta manera se distingue una etapa de aprendizaje o entrenamiento y una etapa de operación o funcionamiento [Hilera, J. R., Martínez, 2000].

Utiliza un aprendizaje no supervisado de tipo hebbiano. Una vez que la red aprendió los pesos, estos permanecerán fijos [Hilera, J. R., Martínez, 2000].

El algoritmo de aprendizaje suele expresarse utilizando una notación matricial. En tal caso podría considerarse una matriz  $W$  de dimensiones  $N \times N$  que representase todos los pesos de la red [Hilera, J. R., Martínez, 2000].

Esta matriz es simétrica, al cumplirse que  $w_{ij}=w_{ji}$  y tiene una diagonal principal con valores nulos debido a la no existencia de conexiones auto-recurrentes ( $w_{ii}=0$ ) [Hilera, J. R., Martínez, 2000].

Las redes Hopfield son muy utilizadas en tareas denominadas auto asociación. [Hilera, J. R., Martínez, 2000].

## 2.2 Red Backpropagation

La red de propagación hacia atrás (BPN) está diseñada para que funcione como red multicapa, con propagación hacia delante, utilizando el modo supervisado de aprendizaje. Este método se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo que determina la salida de la red ante una entrada determinada [Freeman, J. A., Skapura, 1993].

Las redes de propagación hacia atrás poseen la capacidad de auto-adaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. De esta manera cuando están en funcionamiento tienen la capacidad de generalización (facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento) [Ana Bollella, 1997].

### 2.2.1 Estructura y Aprendizaje de la Red Backpropagation

En una red Backpropagation existe una capa de entrada con  $n$  neuronas y una capa de salida con  $m$  neuronas y al menos una capa oculta de neuronas internas [3].

La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a través de las capas hasta llegar a la capa de salida. Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose éstos valores con la salida esperada para obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una propagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera se han modificado los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del que conocíamos su valor de entrada y la salida deseada que debería generar la red ante dicho patrón [Ana Bollella, 1997].

Esta técnica requiere el uso de neuronas cuya función de activación sea continua, y por lo tanto, diferenciable. Generalmente, la función utilizada será del tipo sigmoideal [Ana Bollella, 1997].

Uno de los mayores problemas con este tipo de redes es que no se pueden establecer reglas para determinar el número de neuronas o número de capas de una red para resolver un problema concreto. No obstante el tamaño de las capas, tanto de entrada como de salida, suelen venir determinado por la naturaleza de la aplicación [Ana Bollella, 1997].

## 3 Propuesta de Implementación de una Red Neuronal

El escenario propuesto por la cátedra Inteligencia Artificial consiste en el desarrollo de un software que de soporte a las decisiones de otorgamiento de crédito de una Empresa financiera. El sistema informático deberá aconsejar acorde un modelo de cliente si una persona que solicita un crédito personal es apta, no apta o se encuentra en una situación dudosa para adquirirlo.

Este tipo de desarrollo amerita la implementación de una red neuronal, debido a que no existen relaciones directas entre cada posible configuración del modelo de cliente y su correspondiente perfil (Apto, No Apto, Dudoso), sino mas bien, existen ejemplos de distintos casos en los que a determinados clientes se le han otorgado o no créditos, o bien se requirió otro tipo de análisis mas subjetivo.

El hecho de que no existan relaciones directas entre cada posible configuración del modelo de cliente y su correspondiente perfil se da porque es imposible tener un ejemplo de cada configuración por que son demasiadas o porque pueden aparecer casos especiales de configuración que no fueron tenidos en cuenta.

El modelo de cliente propuesto y utilizado en la actualidad por la Empresa VISAN S.A. para el otorgamiento de créditos personales, y sobre el cual se basa todo el resto del análisis se detalla en la Tabla 1. Dicho modelo parte de la base en que el cliente cumple con los requisitos esenciales y restrictivos impuestos por la financiera (ejemplo: Edad mínima y máxima, ingreso y/o garante, etc.).

**Tabla 1.** Detalle del Modelo de Cliente. Fuente: Silvia Inés Vicentín [VISAN S. A., 2006]

Características	Descripción
Edad	Importante para evaluar el resto de las características según la cantidad de años de la persona.
Nacionalidad	Característica necesaria a la hora de evaluar la historia de un cliente.
Estado Civil	Refleja el grado de responsabilidades a cargo y cultura del cliente.
Personas a Cargo	Implica el compromiso de un egreso patrimonial del cliente.
Situación Laboral	Importante para determinar el grado en el que el cliente puede responder a la cancelación del crédito.
Años de Antigüedad	Refleja la solides del desempeño laboral del cliente, respecto a la edad actual.
Situación en el VERAZ	Refleja el grado de endeudamiento y compromisos financieros con otras empresas.
Ingreso Salarial	Importante para determinar la medida en la que el cliente puede cancelar la cuota con un ingreso mensual propio, y el grado de dependencia hacia un garante.
Inmuebles	Importante para ver el grado en que el cliente puede cancelar la deuda ante alguna irregularidad en cumplimiento del pago del crédito.
Muebles	Ídem Inmuebles, con la diferencia que éste último posee mayor liquides y es más utilizado.
Créditos Cancelados	Refleja la historia financiera y la responsabilidad con la cancelación favorable de los créditos de la Empresa.
Créditos Vigentes	Importante para determinar el grado de compromisos financieros respecto a las características anteriores.

### 3.1 Implementación de una Red Neuronal Hopfield

La primera propuesta de implementación del sistema se realiza a través del modelo de Hopfield. El modelo de cliente definido por la Empresa Financiera privada, esta determinado de tal manera que cada característica del modelo se corresponda con una neurona en Hopfield (45 neuronas). Este número de neuronas en Hopfield se logra dicretizando cada unas de las características del modelo de cliente en cuestión.

### 3.2 Implementación de una Red Neuronal Backpropagation (BPN)

La segunda propuesta de implementación del sistema es a través de la Red de Propagación hacia Atrás (BPN). El modelo de cliente definido por la Empresa Financiera privada, esta definido de tal manera que cada característica se corresponde con una neurona de entrada (12 neuronas), la salida deseada correspondiente a cada patrón de

entrenamiento se corresponde con dos neuronas de salida (10 – apto, 01 – no apto, 11 – dudoso), luego la cantidad de neuronas de la capa oculta se encuentra definida con un total de 67 neuronas mediante prueba y error con un error mínimo relativamente aceptable para el problema.

### 3.3 Arquitectura del Software y Herramientas de Desarrollos

Para ambas implementaciones el lenguaje utilizado es C# con Framework 2.0 bajo el entorno de desarrollo Visual Studio 2005 y Sharp Develop 2.0.

La arquitectura del software cuenta de 3 capas: presentación (interfaz), lógica de negocios (algoritmos implementados) y capa de persistencia (base de entrenamiento para BPN y Hopfield, y las matrices de peso correspondientes al entrenamiento en BPN) [Syed Merhoz Alam, 2004].

#### 3.3.1 Módulos del Sistema

En la Tabla 2 se describen los módulos esenciales del sistema, es decir, los módulos que encierran toda la lógica de la implantación de los dos tipos de redes en estudio.

**Tabla 2.** Descripción de los módulos del sistema.

Módulos	Descripción
Entrenamiento	Módulo que se descompone en dos sub-módulos correspondientes a la red de Hopfield y Backpropagation. El primero se ejecuta al mismo instante en el que se ejecuta el módulo de Reconocimiento correspondiente; mientras que el segundo se lo debe ejecutar de forma independiente desde el cual que se tiene acceso a las configuraciones de la cantidad de neuronas ocultas, factores de aprendizaje y momento, error mínimo y la cantidad de iteraciones máxima.
Reconocimiento	Nuevamente este módulo se descompone en dos, un sub-módulo para cada red. En ambos casos opera siempre sobre su correspondiente matriz de pesos estabilizada/entrenada.
Altas y Bajas de Patrones de Entrenamientos de BPN	Módulo que permite realizar las altas y bajas de los patrones de la base de entrenamiento de la BPN. Esto permite tanto la modificación de la base como la ampliación de la misma para un entrenamiento más robusto.
Explorador de Patrones	Permite ver la configuración de cada patrón de entrenamiento correspondiente a los dos tipos de red.

## **4 Problemas Encontrados y Soluciones**

### **4.1 Obtención del Modelo de Cliente y de los Patrones de Entrenamiento**

La recolección de información necesaria acerca de las características del un modelo de cliente a tener en cuenta para realizar el trabajo resulta un tanto complicada. Esto se debe a que en los bancos, las personas encargadas de decidir acerca de los créditos, se excusan aduciendo no tener tiempo. De esta manera se realizó varios intentos de entrevistas, concurriendo 3 veces al Banco del Chaco y 2 veces al Banco Bisel hasta que se pudo lograr el objetivo.

Las entrevistas no fueron muy productivas ya que las personas consultadas se mostraron reacias a brindar la información necesaria fundamentando que no les es posible hacerlo por políticas internas. La información obtenida que pudo ser aprovechada fue con respecto al Veraz y a los montos de sueldo mínimos que cada banco tenía en cuenta para dar un crédito.

Por último para completar el análisis se logró entrevistar a la persona Responsable de Finanzas de la Financiera VISAN S.A. Srta. Silvia Inés Vicentín, quien brindó información para terminar de comprender el modelo de cliente que la Empresa utiliza en la actualidad y el cual se presentó en este informe.

La Financiera VISAN S.A. brinda créditos prendarios, hipotecarios y personales. Por ello es importante aclarar que sólo se tuvo en cuenta el análisis del modelo de cliente respecto a créditos personales.

El paso siguiente a obtener el modelo de cliente es obtener patrones reales para el entrenamiento de las redes. Esto es aún más difícil de conseguir, debido al gran número que se requiere para el entrenamiento de la BPN.

La escasez de tiempo de la fuente requirió visitas nuevamente a bancos y financieras para solicitar ejemplos de configuraciones, según el modelo obtenido, para casos en los que se podía otorgar el crédito, para casos en los que no y para casos en que se debía realizar un estudio más minucioso de la situación del solicitante. Para contar al fin con ejemplos suficientes para el entrenamiento y pruebas de las redes, se hizo necesario completar la base de entrenamiento con patrones generados por el equipo de desarrollo con un posterior visto bueno por parte de la fuente.

### **4.2 Implementación del modelo de cliente**

Una vez obtenido el modelo de cliente correspondiente, surgió la complicación de llevar esos datos a una forma de presentación que sea consistente entre cada uno de los patrones de ejemplo para el entrenamiento de la BPN. El principal inconveniente fue encontrar la forma de presentar los datos de manera de evitar grandes oscilaciones de los valores en determinadas características entre un patrón y otro (Ej.: patrón1=... $N_i$  = 48000,.....; patrón 2 =... $N_i$  = 0).

La solución implementada para características de ese tipo fue representar mediante índices. A continuación se explica con un ejemplo la solución implementada.

**Características proporcionadas por la Financiera.** Inmuebles = 48000 (representa el valor del inmueble utilizado para evaluar la situación económica del cliente).

**Característica implementada en el software.** Inmuebles = 48 (representa la cantidad de veces el monto del crédito solicitado con la misma información que antes para la toma de decisiones).

En cuanto a la implantación de la red de Hopfield surge el problema a la hora de desglosar ó bien discretizar las características para cada uno de los patrones de entrenamiento, para así obtener la suficiente robustez al momento en que la red deba inferir o clasificar los patrones de entrada distorsionados ó nunca antes vistos. Toda decisión en cuanto a los posibles rangos que puede tomar cada característica siempre fue bajo el consentimiento de la fuente mencionada anteriormente.

### 4.3 Ortogonalidad

El dominio del problema hizo que fuera muy difícil cumplir con la ortogonalidad en la red de Hopfield, por lo tanto las diferencias de configuraciones entre un solicitante y otro que debían tener a la postre distintos perfiles (Apto, No Apto, Deseado) al principio eran muy idénticas entre ellas. Esto hacía muy difícil el aprendizaje correcto de la red, ya que no había la diferencia necesaria entre entradas con distintos perfiles de salida. Debido a ello a la hora de su funcionamiento muchas veces asociaba determinadas entradas con perfiles incorrectos.

La solución a este inconveniente fue el siguiente: a pesar de que por el número de neuronas era posible que la red aprenda cinco patrones, se disminuyó este número a tres de manera de cumplir con la ortogonalidad y obtener una recuperación perfecta.

### 4.4 Número de Neuronas Ocultas y Valores para los Factores

Como se mencionó en el marco teórico, no existe una forma de saber de antemano el número correcto de neuronas ocultas necesarias para el funcionamiento correcto de una red BPN, así como tampoco el valor que deben tomar los factores.

En el transcurso de la implementación de la red sólo se buscaba el número correcto de neuronas ocultas pero los errores causados por los patrones con que la red era entrenada oscilaban considerablemente haciendo imposible que la red aprendiera. Esto llevó a la utilización de los factores de momento y de aprendizaje, y a la aparición de un nuevo inconveniente, encontrar los valores para cada uno de los factores.

Esto requiere de un arduo trabajo de búsqueda por prueba y error, que consume la mayor parte del tiempo de la implementación de una red Backpropagation; ya que por cada número de neuronas con la que se prueba la red, al no convergir la misma, se debe variar los factores hasta comprobar con exactitud que realmente es necesario agregar nuevas neuronas a la red. Al agregar neuronas la maniobra con los factores se repite.

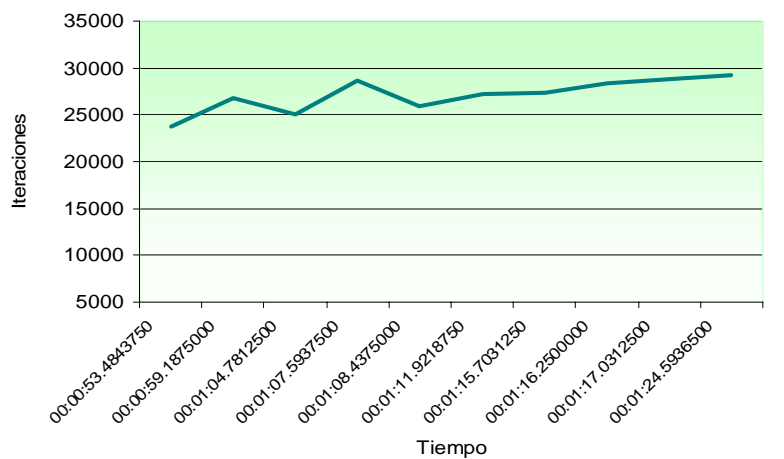
## 5 Conclusión

Luego del análisis e implementación de escenario mediante las redes neuronales Hopfield y BPN, el último paso para cumplir con los objetivos de la comparación de ambos modelos consiste en pruebas sobre el software ya desarrollado, analizando el funcionamiento de cada módulo correspondiente a los modelos de redes en estudio.

A continuación se presentará los resultados de las pruebas realizadas sobre los módulos de entrenamiento y reconocimiento de las redes de Hopfield y BPN.

En cuanto al entrenamiento, se realizaron diez corridas de los módulos correspondientes a cada red. La red de Hopfield lo realiza en forma instantánea al momento previo del reconocimiento de un determinado patrón, es decir el tiempo de entrenamiento es constante de 0 seg. por ser despreciable. Mientras que para la BPN se puede observar la figura 3, donde varía el tiempo de entrenamiento según la inicialización aleatoria de cada una de las corridas. Para dicha prueba se ha utilizado un error mínimo de 0,001.

**Fig. 3.** Tiempos de Entrenamiento en segundos de la Red Backpropagation.

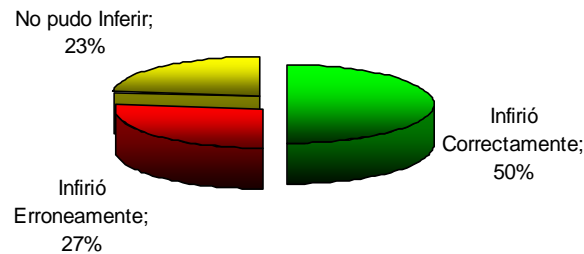


Una vez entrenada la red, los resultados correspondientes al reconocimiento están presentados en gráficos estadísticos del funcionamiento con treinta ejemplos de patrones que no fueron utilizados en el entrenamiento.

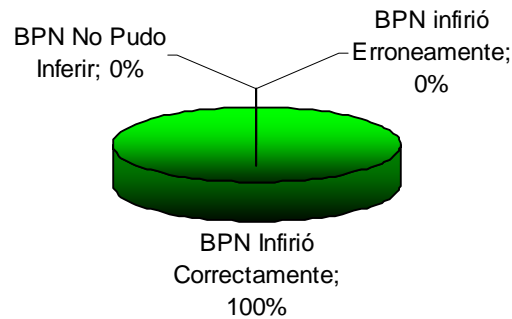
En la figura 1 y 2 se observa los resultados en cuanto a la salida deseada y el funcionamiento correcto de la red de Hopfield y Backpropagation respectivamente.



**Fig. 1.** Resultados de la Red de Hopfield.



**Fig. 2.** Resultados de la Red de Backpropagation.



Como análisis de los resultados de los experimentos se observa en la tabla 3 los principales aspectos a tener en cuenta a la hora de decidir la implementación de un tipo de red ante un escenario determinado.

**Tabla 3.** Aspectos a considerar sobre las redes en estudio.

Aspectos	Hopfield	Backpropagation
Complejidad en la implantación	<i>Simple</i> . Esto es debido al tipo de entrenamiento, una vez hecha la estructura de datos correspondiente solo resta por encontrar una buena configuración de patrones de entrenamiento.	<i>Compleja</i> , ya que se requiere un tiempo considerable debido a la metodología prueba y error para la determinación de la cantidad de capas ocultas, números de neuronas en las mismas y valor de los factores de aprendizaje y momento.
Recursos computacionales	<i>Mínimos</i> . Todo el procesamiento en Hopfield se resume a cálculos matriciales.	<i>Altos</i> , esto es respecto a la fase de entrenamiento, la cual requiere $n$ iteraciones hasta que la red converja. Una vez entrenada la red, el funcionamiento (reconocimiento) puede ejecutarse con recursos computacionales <i>mínimos</i> .

Rendimiento	Según los experimentos realizados, Hopfield cuenta con un rendimiento favorable de tan solo el 50%.	Ante los mismos patrones de pruebas la BPN presento un rendimiento favorable del 100%.
-------------	---	--

Por los resultados expuestos hasta aquí se puede concluir que la implementación de una red Hopfield no es adecuada para este dominio de problema, esto es debido a que es muy difícil cumplir con la ortogonalidad necesaria en los patrones de manera que la red aprenda los suficientes para que en su funcionamiento infiera correctamente las salidas ante cada entrada. Además, el número de neuronas correspondientes al modelo de clientes presentado y las características intrínsecas de este algoritmo limita el número de patrones que la red puede aprender, para una posterior recuperación perfecta de la red en la etapa de funcionamiento según lo esperado por el usuario final en este tipo de escenario.

En cuanto a la ventaja que posee, de ser un algoritmo muy simple de implementar, puede ser adecuada y conveniente en otras situaciones más deterministas. Como por ejemplo: reconocimiento de imágenes y de voz, el control de motores, resolución de problemas de optimización [Hilera, J. R., Martínez, 2000].

En contraposición el algoritmo BPN, a pesar de las desventajas en cuanto al uso recursos computacionales requerido y complejidad en la implementación, se encuadra perfectamente para este tipo de escenarios debido a la robustez de dicha red.

## Referencias

[VISAN S.A, 2006]

Silvia Inés Vicentín, DNI: 26.508.011, Responsable de Finanzas, Río Negro 394 - Goya (Ctes.), Cel: 03777-15534937. Empresa: Horacio Joaquín Vilas - VISAN S.A. Actividad: Financiera de Créditos Prendarios, Hipotecarios y Personales, José E.Gómez 105 - Goya (Ctes.)

[Hilera, J. R., Martínez, 2000]

Hilera, J. R., Martínez, V.J. Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. Alfaomega. 2000.

[Freeman, J. A., Skapura, 1993]

Freeman, J. A., Skapura, D. M. Redes Neuronales. Algoritmos, aplicaciones y técnicas de propagación. Addison-Wesley/ Diaz de Santos. 1993.

[Syed Merhoz Alam, 2004]

Syed Merhoz Alam, Estudiante de Ingeniería en Sistemas de Información, Universidad de NED de Ingeniería y de la Tecnología, Karachi.

<http://www.codeproject.com/csharp/matrix.as>

[Ana Bollella, 1997]

<http://www.monografias.com/trabajos12/redneuro/redneuro.shtml>

<http://www.monografias.com/trabajos12/redneuro/redneuro2.shtml>