

Comparación de Modelos de Redes Neuronales Utilizados en Sistemas de Soporte de Decisiones

Broggi, Carlos Javier
Goujon, Diego Javier
Herrmann, Raúl Alberto

Universidad Tecnológica Nacional, Facultad Regional Resistencia

Abstract

El presente paper realiza una comparación entre las capacidades de dos modelos de redes neuronales artificiales, la red backpropagation y la red de Hopfield, para dar soporte a la toma de decisiones en el ámbito del otorgamiento de créditos bancarios; usando datos fidedignos para el desarrollo. Describe los pasos realizados para el desarrollo de un sistema para este fin, como así también los problemas surgidos y las soluciones implementadas. Expone además los usos recomendados para este tipo de aplicaciones.

1 Redes Neuronales

Las redes neuronales son modelos inspirados en la estructura y funcionamiento de los sistemas nerviosos. Constan de dispositivos elementales de proceso llamados neuronas. La red generalmente se estructura en capas o niveles. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino. Las neuronas están asociadas entre sí por medio de conexiones que tienen asociadas un peso; los valores que tomen estos pesos representan el conocimiento de la red. [1]

1.1 La red backpropagation

Una red backpropagation (BPN) es un tipo de red neuronal que implementa un método que permite a la red aprender la asociación que existe entre los patrones de entrada a la misma y las clases correspondientes. Este método, conocido como backpropagation (propagación del error

hacia atrás) está basado en la generalización de la regla delta. [1]

Este tipo de redes se están aplicando sistemáticamente a distintas clases de problemas, y por ello es difícil hacer una selección de las aplicaciones más relevantes. Esta versatilidad se debe, en gran medida, a la naturaleza de su proceso de aprendizaje. Algunos de los campos generales de aplicación más representativos son los siguientes:

- Codificación de información
- Traducción de texto en lenguaje hablado
- Reconocimiento de lenguaje hablado
- Reconocimiento óptico de caracteres (OCR)
- Aplicaciones cardiológicas, como ser detección de taquicardias ventriculares y supraventriculares y reconocimiento de formas anormales en señales ECG
- Compresión/descompresión de datos

Una característica importante de esta red es la capacidad de generalización, entendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. [1]

La estructura de la red consiste en una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona de una capa está conectada con todas las neuronas de la capa siguiente. [1]

La red utiliza aprendizaje supervisado, por corrección de error off-line.

De forma simplificada, el funcionamiento de la red BPN consiste en un aprendizaje de un conjunto predefinido de pares entrada-salida dados como ejemplo, empleando un ciclo propagación-adaptación de dos fases:

primero se aplica un patrón de entrada como estímulo para la primera capa de neuronas de la red, se va propagando a través de todas las capas hasta generar una salida, se compara el resultado obtenido en las neuronas de salida con las salidas que se desea obtener y se calcula un valor del error para cada neurona de salida. A continuación estos errores se transmiten hacia atrás partiendo de la capa de salida, hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de la neurona intermedia en la salida original. Este proceso se repite capa por capa hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona de manera que en la siguiente vez que se presente el mismo patrón la salida esté más cercana a la deseada; es decir, el error disminuye. [1]

El algoritmo backpropagation utiliza también una función o superficie de error asociada a la red, buscado el estado estable de mínimo error a través del camino descendiente de la superficie del error. Por ello, realimenta el error del sistema para realizar la modificación de los pesos en un valor proporcional al gradiente descendiente de dicha función de error. [1]

1.2 Red de Hopfield

El modelo de Hopfield consiste en una red monocapa con neuronas cuyos valores de salida son binarios. Cada neurona de la red se encuentra conectada a todas las demás (conexiones laterales), pero no consigo misma (no existen conexiones autorrecurrentes). Además, los pesos asociados a las conexiones entre pares de neuronas son simétricos. Las funciones de activación de las neuronas son del tipo escalón. [1]

Se trata de una red autoasociativa. Así, varias informaciones (patrones) diferentes pueden ser almacenadas en la red durante la etapa de aprendizaje. Posteriormente, si

se presenta a la entrada alguna de las informaciones almacenadas, la red evoluciona hasta estabilizarse, ofreciendo entonces en la salida la información almacenada, que coincide con la presentada en la entrada. Si, por el contrario, la información de entrada no coincide con ninguna de las almacenadas, por estar distorsionada o incompleta, la red evoluciona a través de una serie de iteraciones, generando como salida la información almacenada más parecida. Que la red converja depende de que la cantidad de neuronas sea suficiente para almacenar los patrones que se le enseñan y de la ortogonalidad entre éstos. [1]

Utiliza un mecanismo de aprendizaje no supervisado, hebbiano, off-line. El peso de una conexión entre una neurona i y otra j se obtiene mediante el producto de los componentes i -ésimo y j -ésimo del vector que representa la información o patrón que debe almacenar. Si el número de patrones a aprender es M , el valor definitivo de cada uno de los pesos se obtiene mediante la suma de los M productos obtenidos mediante el procedimiento anterior. [1]

2 Escenario

Una entidad bancaria necesita un sistema que le dé soporte a la toma de decisiones en el sector de créditos. Se desea utilizar la información disponible sobre ciertas características de un cliente para determinar si se le otorga o no el crédito.

La base de datos utilizada como referencia¹ contiene las siguientes características:

- Estado de la cuenta corriente

¹ La base de datos tomada como referencia para realizar el sistema fue obtenida del UCI Repository Of Machine Learning Databases and Domain Theories. Forma parte de las bases de datos Statlog, que contienen información sobre el mundo real para ser utilizada en algoritmos de aprendizaje de máquina, estadísticos y redes neuronales. Estas bases de datos son un subconjunto de las bases de datos utilizadas por el European StatLog Project. La base de datos utilizada fue provista por el profesor Dr. Hans Hofmann, del Institut für Statistik und Ökonometrie, Universität Hamburg, FB Wirtschaftswissenschaften. (Instituto de Estadística y Econometría, Universidad de Hamburgo, Área de Ciencias Económicas).

- Duración del crédito en meses
- Historia de créditos
- Propósito
- Monto del crédito
- Cuenta de ahorro/bonos
- Tiempo en el empleo actual
- Porcentaje del ingreso libre disponible utilizado para el pago de la cuota
- Estado civil y sexo
- Existencia de co-solicitante/garante
- Tiempo en la residencia actual
- Propiedades
- Edad en años
- Otros planes de financiamiento
- Vivienda
- Número de créditos existentes en este banco
- Trabajo
- Número de personas a las que tiene que mantener
- Teléfono
- Trabajador extranjero al banco

Según las características que presenta cada cliente se lo clasifica como apto o no apto para el crédito.

3 Propuesta planteada

Se desarrollaron e implementaron un modelo de red de Hopfield y un modelo de red backpropagation para dar soporte a la toma de decisiones en el otorgamiento de créditos bancarios.

3.1 Modelo de Hopfield

Implementación del software

Dado que no se sabía a priori la codificación que se utilizaría, y por lo tanto el número de neuronas que debía poseer la red, el software implementado crea redes de Hopfield con diferentes cantidades de neuronas de acuerdo a la cantidad de columnas que posee la información de entrada. Esto permitió realizar diversas pruebas con distintas codificaciones. El software

admite cargar un conjunto de prueba y comparar la salida de la red con las salidas esperadas, esto permite evaluar el rendimiento de la red para cada tipo de codificación y conjunto de prueba utilizados.

Codificación de la información

Dado que el modelo de Hopfield implementado acepta únicamente entradas binarias, el primer paso fue codificar los valores de cada campo para cumplir con esta restricción.

La primera codificación se realizó de la siguiente manera: para las características no numéricas se utilizó una neurona por cada valor posible de la característica, de manera que un 1 en la primera neurona correspondiente a una característica (el resto de las neuronas para esa característica toman valores de -1) represente el primer valor que puede tomar dicha característica, un 1 en la segunda neurona de la característica representa el segundo valor posible de la característica y así sucesivamente.

Los valores numéricos se dividieron en rangos, procediéndose luego de la misma manera que para los valores no numéricos (una neurona por cada rango de la característica).

Las pruebas realizadas con los valores codificados de esta manera fueron poco satisfactorias. Se entrenó a la red con diferentes conjuntos de patrones. Los patrones en dichos conjuntos alcanzaban el 30% pero no el 50% de ortogonalidad.

Estos patrones resultaron ser muy poco representativos del total de patrones y la red siempre convergía a un mismo patrón.

Dadas las dificultades arriba mencionadas que presentaba esta codificación, se procedió a realizar una totalmente diferente.

La nueva codificación consistió en asignar tres neuronas por característica (excepto para las columnas que ya poseen valores binarios y de poca influencia, que se codifican con dos neuronas, asignando 1, -1 a un valor y -1, 1 al otro). Se asignó a los valores más favorables (obtenidos en base a estadísticas) la codificación 1, 1, 1, y a los más desfavorable la codificación -1, -1, -1. A los valores

intermedios se le asignó la codificación 1, 1, -1 o -1, -1, 1, de acuerdo a si tendían a ser más favorables o desfavorables.

Se enseñó a la red de distintas maneras. Primero se le enseñó con dos patrones: el que representa al mejor cliente de la base de datos y el que representa al peor. Luego se le enseñó con conjuntos de los mejores y peores clientes. Por último se enseñó a la red con dos patrones ficticios: uno que representa a un cliente ideal que posee todas las características favorables y otro que representa a un cliente que posee todas las características desfavorables. Lo que se intentó con este último conjunto de entrenamiento es enseñar a la red patrones libres de ruido, considerando que los ejemplos de la base de datos presentan todos cierto nivel de distorsión.

Los mejores resultados se obtuvieron con el último conjunto de aprendizaje.

3.2 Modelo de red backpropagation

Implementación del Software

Dado la imposibilidad de determinar a priori la óptima estructura de la red, se decidió implementar un software que permita variar los parámetros de la red. Este software permite especificar: la cantidad de capas de la red, la cantidad de neuronas de cada capa, la tasa de aprendizaje, el factor de momento, el tipo de función de activación (sigmoideal o simétrica logarítmica²) y el uso o no de neuronas de tendencia. Además permite parar el aprendizaje luego de una cierta cantidad de iteraciones, o cuando se haya llegado a un error global o por patrón predefinido.

Se muestra en tiempo real el error global y el máximo error por patrón, lo que permite tener conocimiento sobre como evoluciona la red.

También es posible cargar un conjunto de patrones de prueba y observar la salida de la red junto con la salida deseada para

cada uno de ellos, como así también la cantidad de aciertos y errores. Asimismo se identifican los patrones que fueron clasificados en forma errónea.

Codificación de la información

La base de datos original contiene algunos campos con contenido no numérico. Dado que la red no acepta este tipo de datos, se procedió a codificarlos en forma numérica.

Para realizar la codificación de cada una de estas características, se realizó un análisis estadístico de las mismas, determinando la probabilidad de que se obtenga el crédito dado un valor de característica. En función a esto se las codificó en un rango de -5 a 5, correspondiendo -5 al valor más desfavorable y 5 al valor más favorable de la característica.

Inicialmente las columnas con valores numéricos no se codificaron, manteniendo sus valores originales.

Las primeras pruebas, realizadas con este conjunto de datos, arrojaron resultados muy desfavorables. La red producía valores de salida muy similares y hasta idénticos para cada una de las entradas. Se variaron todos los parámetros de la red y se llevó a cabo un gran número de pruebas, obtenido exactamente los mismos resultados. Haciendo un análisis de los pesos de las conexiones y de las salidas se llegó a la conclusión de que el estado de cada neurona estaba en función de casi exclusivamente una sola entrada. Esta entrada contenía valores en un rango de 250 a 18.500, estos valores eran extremadamente grandes en comparación con la mayoría de las demás entradas que tenían valores mucho más pequeños.

Para verificar que el problema era debido a los grandes valores de ciertas columnas, se entrenó a la red con patrones que no contenían esas columnas. Acertadamente, el problema anterior no se presentaba con estos patrones.

Para solucionar el problema de los valores grandes, se procedió a escalar las columnas que contenían dichos valores: monto del crédito, duración del crédito y edad del solicitante.

² Obtenida de la ayuda del software de prueba NNDT (Neural Network Development Tool)

Utilizando esta nueva codificación en los ejemplos de aprendizaje los resultados mejoraron significativamente

Con el mismo criterio, para lograr una mayor uniformidad en los valores, se los normalizó. También se probó escalar de diferentes maneras los valores de las columnas grandes. Sin embargo, las pruebas realizadas sobre los patrones codificados de estas maneras arrojaron resultados peores que los anteriores, por lo que se decidió mantener la codificación previa.

Estructura de la red

En base a las pruebas realizadas variando la cantidad de capas y neuronas, se llegó a las siguientes conclusiones: para una capa oculta con menos de 30 neuronas el error global es muy alto y se producen cambios bruscos en él durante el aprendizaje; entre 30 y 70 neuronas el error global descende más uniformemente, pero la red no logra generalizar y sobre el conjunto de prueba se obtienen resultados poco satisfactorios; con una cantidad aproximada de 80 neuronas se obtuvieron los mejores resultados en tiempos de aprendizaje aceptables (menores a un minuto); con una mayor cantidad de neuronas o más capas, el tiempo de aprendizaje aumenta notablemente y no se notan mejoras en los resultados.

Respecto a la cantidad de iteraciones, se observa que el error global descende muy rápidamente en las primeras iteraciones. A partir de las 200 iteraciones el error global decrece muy lentamente. Por lo tanto, se limitó el aprendizaje a las 300 iteraciones.

Se probaron dos funciones de activación para las neuronas de la capa oculta: la función sigmoideal entre 0 y 1, y la función simétrica logarítmica. Las neuronas de entrada usan siempre la función de activación identidad, y las de salida la función sigmoideal entre 0 y 1. La función logarítmica resulta mucho más sensible al valor de la tasa de aprendizaje y produce oscilaciones excepto que ésta sea muy pequeña (menor a 0,01). La función sigmoideal produce mejores resultados y un tiempo de aprendizaje menor. Con tasas de aprendizaje mayores a

0,4 esta función también produce oscilaciones. Los mejores resultados se obtuvieron con tasas de 0,15 a 0,20.

Con la utilización de neuronas de tendencia y factores de momento, no se encontraron mejoras significativas en los resultados que justifiquen la sobrecarga de procesamiento.

Selección de los conjuntos de entrenamiento y prueba

La base de datos cuenta con 1000 ejemplos. Estos ejemplos se dividieron en dos conjuntos mutuamente excluyentes: un conjunto de aprendizaje y otro conjunto de prueba. En diferentes pruebas, se hizo variar el tamaño del conjunto de aprendizaje entre un 70% y un 90% del total de ejemplos disponibles, dejando los ejemplos restantes para el conjunto de prueba.

Se entrenaba a la red con el conjunto de aprendizaje y se realizaban pruebas con los ejemplos restantes.

Luego de diversas pruebas con distintas configuraciones de la red y tamaños de conjuntos de aprendizaje, se notó que existían ciertos patrones del conjunto de prueba que eran clasificados erróneamente por la red en forma recurrente.

Se llegó a la conclusión de que no se le estaba enseñando a la red un conjunto representativo de patrones.

Para conseguir un conjunto de aprendizaje que fuera lo suficientemente representativo del total de patrones, se implementó una funcionalidad que identifica los patrones que son clasificados erróneamente y los incorpora al conjunto de aprendizaje, pasando a su vez patrones del conjunto de aprendizaje al conjunto de prueba, para mantener la proporción de ambos conjuntos.

Es decir, se intenta enseñar a la red la mayor cantidad de patrones diferentes posibles.

Una vez conseguido un conjunto de entrenamiento lo suficientemente representativo, se enseñaron estos ejemplos a una red nueva y se verificó la cantidad de aciertos y errores sobre el nuevo conjunto de prueba que contenía los patrones que la red no conocía.

De esta manera se logró incrementar el porcentaje de aciertos de un 65% a un 90 y hasta 95%.

4 Comparación de los Modelos

Analizando los resultados obtenidos de las pruebas realizadas a ambos modelos, se llegó a las siguientes conclusiones.

El modelo de red BPN presenta un mejor rendimiento que el modelo de Hopfield. Con el primer modelo se obtuvo un porcentaje de aciertos de hasta 95%. En cambio, con el modelo de Hopfield no se logró superar el 77% de aciertos.

Estos resultados probablemente se deban a que la capacidad de aprendizaje de una red BPN es superior a la de una red Hopfield, debido a las restricciones de la última y la mayor flexibilidad de la primera. Es probable que existan ejemplos en este dominio de problema en particular cuyos valores de características difieran muy poco, pero tengan salidas distintas. Lo cual excede las capacidades de Hopfield.

El tiempo de aprendizaje es mínimo en Hopfield, no tan así en BPN.

Hopfield es un modelo más rígido, ya que no existen parámetros que puedan ser modificados. En cambio, una red BPN admite la variación del número de capas, neuronas y otros parámetros, lo que permite un mejor ajuste a un dominio de problema particular, es decir es más flexible. Si bien esto es una ventaja, demanda un tiempo de prueba mayor.

Sin embargo una vez que las redes aprendieron, ambas presentan los resultados casi en forma instantánea.

El tiempo de desarrollo es bastante mayor para el caso de una red BPN en contraste con el tiempo de desarrollo de una red Hopfield; esto se debe a que la estructura misma de esta última es más simple.

También es más difícil la codificación de la información en el caso de una red Hopfield por admitir solamente valores binarios, en el caso de una red BPN los valo-

res no tuvieron que sufrir cambios tan importantes, la codificación fue más directa.

5 Conclusiones

La utilización de modelos de redes neuronales para dar soporte a la toma de decisiones es posible.

Recomendamos que la utilizada sea una red BPN, no una red Hopfield debido a las mejores prestaciones de la primera.

Una aplicación de este tipo debe ser considerada como una herramienta más para ser utilizada por la persona encargada de tomar decisiones, no es posible delegar la toma de decisiones completamente a la aplicación, ya que existen factores subjetivos e implícitos que inciden en el otorgamiento o no de un crédito y que no pueden ser modelados.

Un contraste entre la decisión del encargado y la salida del sistema puede considerarse un indicador de que debe realizarse un análisis más profundo del caso.

Dado el tiempo relativamente corto que emplea la red para aprender, es recomendable reentrenarla periódicamente, lo cual permite reflejar las políticas actuales.

Referencias

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

Datos de contacto

Carlos Javier Broggi
Av. Castelli 659 Piso2 Dpto. 5A
Resistencia, Chaco, Argentina
carlosjbroggi@gmail.com

Diego Javier Goujon
Av. Belgrano 151 5° B
Resistencia, Chaco, Argentina
diegogoujon2@hotmail.com

Raúl Alberto Herrmann
Av. Castelli 659 Piso2 Dpto. 6A
Resistencia, Chaco, Argentina
raulherrmann@gmail.com