

SISTEMA DE RECONOCIMIENTO DE DÍGITOS MANUSCRITOS UTILIZANDO REDES NEURONALES

Juan Manuel Martínez Cabrera

Universidad Autónoma de Zacatecas
klamath135@yahoo.com

Aldonso Becerra Sánchez

Universidad Autónoma de Zacatecas
a7donso@uaz.edu.mx

Gustavo Zepeda Valles

Universidad Autónoma de Zacatecas
gzepeda@hotmail.com

Santiago Esparza Guerrero

Universidad Autónoma de Zacatecas
chagouaz@hotmail.com

Uriel Ramírez García Correa

Universidad Autónoma de Zacatecas
uramirezgc@uaz.edu.mx

Resumen

El reconocimiento de dígitos manuscritos es un entorno en creciente uso y por consiguiente requiere ir abordando alternativas para su implementación, el uso de redes neuronales ha venido retomando el auge dentro del área de reconocimiento de patrones. Este documento muestra el uso de redes neuronales, a través de un software personalizado, como el motor detrás un sistema de reconocimiento de caracteres ópticos. En este sistema los dígitos numéricos son simplificados a través de filtros de imagen y luego presentados como entrada a la red neuronal para entrenarla (usando el algoritmo de retro-propagación) y ser capaz de clasificar otras muestras en la etapa de pruebas. Los resultados muestran tasas

de reconocimiento cercanas al 85%, que se pueden considerar como aceptables para topologías de una sola capa, dejando pendiente para futuros experimentos el trabajo con redes multicapa pre-entrenadas, ya que suelen incrementar fuertemente su eficiencia.

Palabra(s) Clave(s): Aprendizaje de máquina, perceptrón multicapa, reconocimiento de caracteres ópticos.

1. Introducción

El OCR o Reconocimiento Óptico de Caracteres es la identificación de caracteres impresos mediante el uso de dispositivos fotoeléctricos y software. Si bien el trabajo con los OCR no es nuevo y existen varios métodos para la clasificación de los patrones existen aún rutas que resolver para llegar a poder aplicarlos con mayor eficacia. Algunos de los métodos tradicionalmente usados se basan en los algoritmos de k-vecinos, las maquinas de soporte vectorial (SVM) y comparación contra plantillas (Template Matching). En este sentido existen trabajos que han manifestado el interés y desarrollo de propuestas en sistemas de reconocimiento basándose en el método KNN, Geometric features y la base de datos de Brasil PS con 190 muestras de prueba [18]. Wang [19] obtuvo buenos resultados también utilizando el método de SVM y la base de datos de Bangladesh PS con 16,000 muestras. De igual manera, Hull et al. [20] obtuvieron tasas de reconocimiento bastante considerables con el método de Template Matching más el clasificador estadístico mixto y la base de datos de USPS con 9,754 muestras. El interés de utilizar las redes neuronales en el OCR recientemente ha sido debido a su potencial considerando la manipulación libre de topología que la integrará, y que varios trabajos [1 - 3] han evidenciado.

La motivación para la elaboración del presente trabajo es el interés sobre la versatilidad de las redes neuronales convencionales en una gran diversidad de ámbitos y sobre su capacidad para resolver problemas de alta complejidad sin necesidad de conocer su solución, con el objetivo de implementar una red neuronal que se podrá adaptar a otros proyectos en el futuro. Sin embargo, este trabajo se enfoca en adaptar una red neuronal convencional y de uso general para

su utilización en la fase de reconocimiento del sistema OCR a desarrollar, haciéndose pruebas con varias topologías de red para encontrar la más adecuada al reconocimiento de los dígitos numéricos; para llevar a cabo este cometido, se desarrolló un software que permite acoplar las diferentes fases de la implementación de clasificadores de patrones usando redes neuronales, aunque este software tiene flexibilidad de permitir crecer su funcionalidad y en dado caso poder usar otras variantes, tanto para el módulo del clasificador como para los datos de entrada.

Los resultados en los experimentos con diferentes topologías muestran que las tasas alcanzadas, cercanas al 85%, se pueden considerar como aceptables, tomando en cuenta que se usan redes de una sola capa y con pesos aleatorios. Sin embargo es importante mencionar que aún existe trayecto que andar en la experimentación de redes neuronales, pero con multicapa y con pesos pre-entrenados.

El resto del trabajo está compuesto por las siguientes secciones. En la sección 2 se presenta la teoría del OCR, sus pasos y la teoría sobre las redes neuronales. La sección 3 describe la metodología y los materiales. La sección 4 describe el diseño e implementación del sistema. La sección 5 describe los resultados del entrenamiento y decodificación que se le hizo a varias topologías de red. La sección 6 presenta las conclusiones y trabajos futuros.

2. Principios de reconocimiento óptico de caracteres

Etapas de un OCR

Todos los sistemas de OCR siguen una serie de etapas básicas para alcanzar su cometido [4 - 9]. Estas etapas en general son:

- **Obtención de Imagen** – En esta etapa hay dos formas en cómo se obtiene los datos de entrada al sistema, el modo “Offline” y el modo “Online” [5]. En el modo Offline se trabaja con imágenes digitales de un escáner, una cámara u otra fuente digital que contienen texto a reconocer. En el modo Online se trabaja con datos obtenidos en tiempo real de un dispositivo. Estos datos son información vectorial que describen trazos.

- **Segmentación** – En esta etapa se busca extraer cada caracter de la imagen. La etapa de segmentación es una parte integral para cualquier sistema de reconocimiento de texto [8]. Divide la imagen en diferentes caracteres aislados que permite clasificarlos individualmente, de esta forma asegura la eficiencia de la clasificación.
- **Pre-Procesamiento** – Esta etapa sirve para limpiar, normalizar y simplificar la imagen para que la siguiente etapa pueda realizar su tarea de forma eficaz y eficiente [10]. Algunos de estos procesos son:
 - ✓ Auto-Cropping en donde se le remueve espacio en blanco alrededor del caracter.
 - ✓ Normalización en donde la imagen es transformada a un tamaño estándar.
 - ✓ Binarización en donde la imagen es simplificada en un mapa de bits (bitmap) de 0 y 1s,
 - ✓ Los filtros que semuestran en la figura 1 son los de auto-cropping, normalización y binarización.

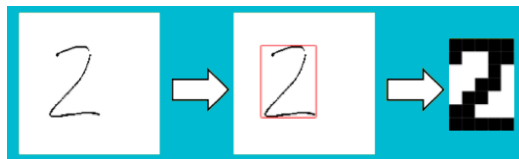


Figura 1 Pre-procesamiento de Imagen.

- **Entrenamiento / Clasificación** – El objetivo de esta última etapa es el de asociar el conjunto de datos obtenidos en la etapa anterior a un caracter de una clase en particular. Existen motores para lograr este fin, algunos son: El método K-NN (K Vecinos), Template Matching (Comparar contra una Plantilla), los Support Vector Machines y las Redes Neuronales Artificiales.

Redes Neuronales Artificiales

Una red neuronal artificial consiste de un número de procesadores muy simples e interconectados llamados neuronas, los cuales son análogos a las neuronas biológicas en un cerebro. La ventaja principal de las redes neuronales artificiales

está en su capacidad en aprender automáticamente de ejemplos y tienen buen desempeño con datos con ruido [11]. Las neuronas están interconectadas con conexiones llamadas sinapsis, cada sinapsis tiene un peso numérico asociado a ella. Los pesos son la forma básica de representar la memoria de una red y expresan la potencia de cada neurona. La red neuronal artificial puede aprender a través de continuamente ajustar estos pesos [12].

El autor José Juan Castaneyra-Matus [4] logra tener un porcentaje de reconocimiento del 86.44% con una topología de 35x15x10 con 15,000 muestras de la base de datos MNIST y tasa de aprendizaje de 0.1 en su implementación de reconociendo de dígitos con una red neuronal, y un porcentaje de 86.80 con una topología de 35x20x15x10.

La autora Leticia María Seijas [15] utiliza tres diferentes bases de datos con los dígitos manuscritos, ello no les aplica filtros a las imágenes. Con la base de datos CENPARMI y una topología de 256x160x10 logra un porcentaje de reconocimiento de 89.05%. Con la BD MNIST, topología de 784x100x10 logra un porcentaje de 76.86%. Con la base de datos MNIST binarizada y topología de 784x160x10 logra un porcentaje de 96.22%.

Modelo de una Neurona Artificial

Una neurona artificial es una unidad de procesamiento de información fundamental a la operación de una red neuronal artificial. En las redes neuronales artificiales, la neurona cuenta con tres elementos básicos de su modelo, sus sinapsis, el sumador y la función de activación [3-9]. Un modelo de una neurona artificial es mostrado en la figura 2.

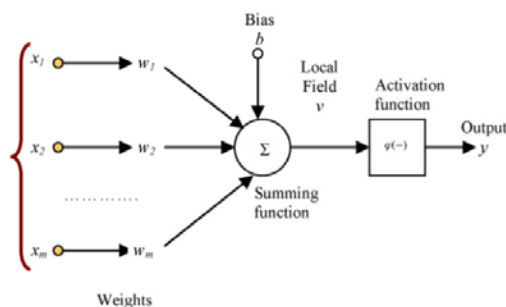


Figura 2 Neurona Artificial.

La neurona cuenta con un conjunto de sinapsis o conexiones de entrada a otras neuronas. Cada sinapsis cuenta con un peso. Los pesos de una sinapsis pueden ir de lo negativo a lo positivo. El sumador se encarga de sumar las señales de entrada multiplicadas por el peso de la sinapsis de la neurona. Este tipo de sumador es un combinador lineal. La función de activación se requiere para limitar la salida de la neurona. El bias b es un parámetro artificial y permite a una red neuronal aprender, aunque todas sus señales de entrada sean ceros. El bias puede tener un valor positivo o negativo.

La neurona artificial cuenta con el sumador (ecuación 1) para procesar sus señales y pesos de entrada. Esta función: k se refiere a la neurona, x_1, x_2, \dots, x_m son las señales de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ son los pesos sinápticos de la neurona k ; v_k es la salida del sumador, b_k es el bias (umbral), y y_k es la salida de la neurona.

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (1)$$

La función de activación (ecuación 2) toma como entrada v_k y produce como salida y_k .

$$y_k = \varphi(v_k) \quad (2)$$

La función Sigmoidea (ecuación 3) es una función de activación concreta, es una de las más populares en la construcción de redes neuronales debido a sus propiedades de ser limitador suave [1]. Mientras que la función escalón toma el valor de 0 o 1, la función sigmoidea toma un rango continuo de valores de 0 a 1.

$$\varphi(v) = \frac{1}{1+e^{-v}} \quad (3)$$

Aprendizaje Supervisado

En el entrenamiento supervisado la red neuronal es entrenada con los datos de entrada que llevan asociados sus respectivas salidas esperadas. Durante el entrenamiento se calcula la salida correspondiente a un conjunto de datos de entrada dado y se compara con la salida esperada, de esta forma es posible conocer el error y ajustar los pesos de acuerdo a este [4] [13].

Arquitectura de Red Alimentación Hacia Delante

En una red neuronal de alimentación hacia delante (Feed Forward) las neuronas se organizan en forma de niveles llamadas capas. Las neuronas de una capa se conectan con todas las neuronas de la capa siguiente y de la capa anterior en la red. El flujo de datos introducidos es estrictamente hacia delante, los datos pasan desde los nodos de entrada hasta los de salida [5].

Son redes no cíclicas, las salidas de una capa no afectan esa misma capa [14]. Una vez que esté entrenada una red neuronal Feed-Forward, su estado es fijo y no se altera cuando se le presentan nuevos datos de entrada. La figura 3 muestra esta topología.

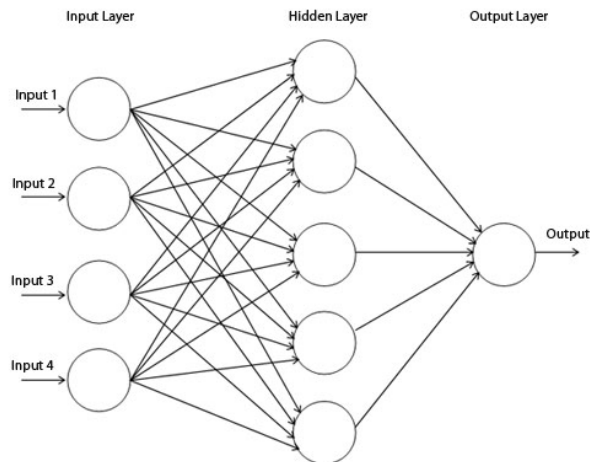


Figura 3 Perceptrón Multicapa con Alimentación Hacia Delante.

Red Neuronal Perceptrón Multicapa

El perceptrón multicapa (MLP–Multi Layered Perceptron) es una extensión al perceptrón simple y es uno de los tipos de red neuronal más conocido y utilizado, esta red se muestra en la figura 3. El perceptrón multicapa tiene una o más capas entre las capas de entrada y salida, estas capas que no están directamente conectadas al entorno son llamadas ocultas [12]. Este tipo de red es de aprendizaje supervisado y es de alimentación adelante, utiliza el algoritmo de retro propagación con deseno gradiente para su aprendizaje. Debido a su estructura

extendida, este perceptrón puede resolver cualquier operación lógica, y generalizar cualquier tipo de función.

Algoritmo de Retro propagación con Descenso de Gradiente

El algoritmo de retro-propagación es uno de los más conocidos e implementados para los perceptrones multicapa [12]. El algoritmo utiliza la técnica de descenso por gradiente para encontrar el mínimo de la función de costo que mide el error del sistema como una función diferenciable de los pesos. El enfoque estocástico tiene la ventaja de permitir una amplia exploración de la superficie de costo. Los patrones de entrenamiento son presentados a la red en orden aleatorio, modificándose los pesos luego de presentar cada patrón [15]. Se basa en minimizar la diferencia entre la salida deseada y la salida real.

Entrenando la Red Neuronal

Los siguientes pasos describen la forma en que una red neuronal aprende:

- Se inicia la red neuronal con valores aleatorios para los pesos sinápticos, valores decimales del 0 a 1.
- Presentarle a la red la imagen en forma de vector y propagar la actividad de forma hacia delante desde la capa de inicio a las capas ocultas y finalmente a la capa de salida.
- Calcular el valor de error (E) de cada neurona en la capa de salida. La salida (Y) de la neurona menos el valor deseado (D) ecuación 4.

$$E_i = Y_i - D_i \quad (4)$$

- Calcular el valor delta Δ de cada neurona en la capa de salida ecuación 5, este para actualizar los pesos sinápticos a esta neurona.

$$\Delta_i = Y_i (1 - Y_i) E_i \quad (5)$$

- Se retro-propagan los valores delta de las neuronas en la capa de salida a las capas ocultas para calcular sus propios valores delta ecuación 6.

$$\Delta_i = Y_i (1 - Y_i) \sum Y_k W_k \quad (6)$$

- Se actualizan el bias ecuación 7 y los pesos sinápticos ecuación 8 de todas las neuronas. Donde ϵ es la tasa de aprendizaje y X se refiere a la salida de la neurona emisora a esta o sea una señal de entrada.

$$b_{i+1} = b_i + \epsilon \Delta \quad (7)$$

$$W_{i+1} = W_i + \epsilon X \Delta_i \quad (8)$$

- Se calcula el error promedio cuadrado MSE para esta imagen ecuación 9.

$$mse = \frac{1}{2} \sum_{i=1}^n (Y_i - D_i)^2 \quad (9)$$

- Repetir los pasos del 2 al 6 hasta agotar todas las imágenes en el conjunto de entrenamiento. Con esto termina una época y se calcula el MSE promedio de la época.
- Si el MSE promedio de la época es menor al umbral definido por el usuario el entrenamiento termina, por lo contrario se inicia otra época con el paso 2.

3. Metodología

Esta sección describe los pasos que se siguen en el proceso de desarrollo del trabajo presentado.

Proceso General

Los siguientes pasos definen el proceso general del proyecto:

- Se hizo la adquisición del corpus de datos. Se pidió a personas diversas que proporcionaran muestras de dígitos manuscritos. Esto con el fin de construir un corpus con suficiente variedad.
- Se separó el corpus de datos en dos conjuntos. El conjunto de entrenamiento que corresponden al 90% del corpus de datos y el conjunto de pruebas que corresponde al 10% del corpus.
- Se aisló cada dígito de las hojas y se les aplicó procesamiento de imagen de forma individual con los filtros de cropping, redimensión, y simplificación (binarización). Cada imagen de dígito contiene un metadato que dicta que dígito representa para los fines del entrenamiento.

- Se propusieron varias configuraciones de topología de red para encontrar la más óptima al problema de reconocimiento de dígitos manuscritos.
- Se entrenó cada red neuronal con su configuración de topología con el algoritmo de retro-propagación, dándole el conjunto de entrenamiento como su entrada.
- A cada red neuronal entrenada se le aplicó la decodificación o prueba con el conjunto de datos de prueba donde se recolectó el porcentaje de reconocimiento de cada topología.

Muestras de Dígitos Manuscritos

Para que la red neuronal pueda clasificar a un dígito, primeramente, se tiene que entrenar la red con varias muestras que representen a todo el conjunto de dígitos que se van a clasificar. En este sistema se busca poder clasificar a los dígitos manuscritos del cero al nueve y por tanto se requiere de un conjunto de muestras de dígitos lo suficientemente grande para poder reconocer las variaciones en la forma de escribir de la gente y las mismas variaciones en la forma de representar un dígito. En este proyecto se obtienen las imágenes de entrada de modo offline.

Se decidió recolectar el conjunto de muestras de dígitos a mano. Se les pidió a sesenta cinco personas que escribieran a mano dos renglones con los dígitos del cero al nueve, estas personas fueron alumnos de dos salones. En total se recolectaron las diez hojas de trece renglones cada una, cada renglón teniendo los dígitos del cero al nueve, esto dio un total de 1300 muestras o 130 muestras de cada dígito. La figura 4 muestra una de las hojas que se recolectó con varias de las variaciones de los dígitos de diferentes personas.

Las hojas con los dígitos manuscritos fueron escaneadas, dando como resultado un conjunto de imágenes de 1664 x 2163 píxeles. Estas imágenes se convirtieron a escala de gris con el programa Photoshop y luego se les aplicó un threshold de 190, esto para reducir el ruido del escaneo y para limitar los grises a blanco o negro. Esto con el fin de dejar las imágenes con el conjunto de muestras listas para el uso del sistema de este proyecto.



Figura 4. Hoja con Dígitos Manuscritos.

4. Desarrollo

Diseño e Implementación del Sistema de OCR con Red Neuronal MLP

El sistema de OCR se diseñó e implementó bajo el paradigma de orientación a objetos (OOP), con este enfoque cada neurona es un objeto, la red es un objeto, cada capa es un objeto, etc. La implementación se realizó en el lenguaje de programación Java y el interfaz gráfico con JavaFX.

En este sistema se utilizan los filtros de auto-cropping, normalización y binarización para el procesamiento de las imágenes que contienen el dígito a reconocer. El último filtro resulta en una imagen simple de tamaño 5x7 píxeles. El desarrollo de los filtros de imagen no estuvo dentro del alcance de este proyecto y por tanto estos filtros no se desarrollaron en este proyecto y fueron utilizados de una librería por Jeff Heaton [17].

Este sistema toma estas imágenes ya filtradas como su entrada en la fase de entrenamiento y decodificación. La misma red se encarga de la extracción de características.

Las neuronas de la red inician con valores aleatorios para sus pesos de sinapsis y bias, utilizan la función Sigmoidea como su función de activación. Las neuronas cuentan con dos arreglos, uno contiene las referencias a neuronas en la capa anterior, el otro arreglo las referencias a las neuronas en la capa que sigue. Para generar los valores aleatorios de los pesos, el valor inicial, se utiliza el Método de Congruencia Lineal [16] como se muestra en la (ecuación 4), está siendo una

implementación de la librería de Java. En este método x es el valor inicial o semilla, m es un módulo elegido, a y c son valores fijos, ecuación 10.

$$x_{n+1} = ax_n + c(\text{mod } m) \quad (10)$$

Sistema Desarrollado

El interfaz gráfico GUI del sistema cuenta con varias pestañas que le permite al usuario realizar varias operaciones. El programa desarrollado le permite al usuario crear una red neuronal con la configuración que desee, claro solo configuraciones que pertenecen al perceptrón multicapa y hay una pestaña que muestra una visualización de ella, como se muestra en la figura 5.

El sistema cuenta con pestañas para: ver los pesos de las sinapsis antes y después del entrenamiento, una grafo que muestra cómo se entrenó la red neuronal figura 7, entrenar y probar la red neuronal con datos binarios figura 6, entrenar y probar la red con imágenes, y una pestaña que le permite aplicar filtros a las imágenes en forma de grupo (Batch Processing) figura 8.

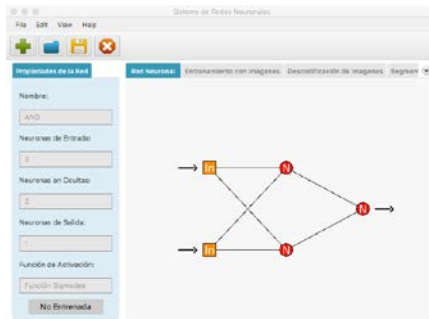


Figura 5 Visualizar la Red.



Figura 6 Entrenar la Red.

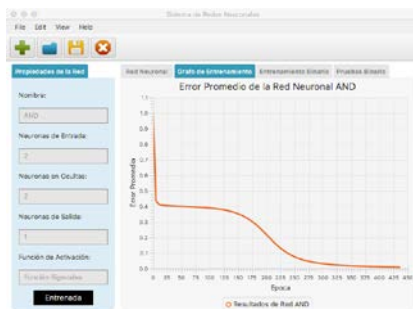


Figura 7 Resultado del Entrenamiento.



Figura 8 Entrenar con Imágenes.

El sistema fue diseñado e implementado primeramente para el reconocimiento de caracteres con una red neuronal como su motor, pero la flexibilidad del sistema es grande. Una ventaja del sistema es que se puede aplicar la red neuronal a otros usos. La red se puede entrenar para aprender funciones lógicas como la del AND, OR, XOR y Negación entre otras desde el mismo interfaz del sistema.

5. Resultados

El corpus de datos contiene un total de 1,300 muestras de dígitos manuscritos. Cada imagen es de 5x7 pixeles y se representa como un vector de 35 bytes. Cada topología de red cuenta con 35 neuronas en la capa de entrada, cada una corresponde a un byte del vector de entrada.

Cada topología de red también cuenta con 10 neuronas en la capa de salida, debido a que hay 10 dígitos o clases que se buscan clasificar. Para cada dígito le corresponde un patrón de salida, este patrón de salida especifica cual neurona debe estar encendida y cuales apagadas. La tabla 1 muestra estos patrones.

Tabla 1 Patrones de Salida.

Dígito	Patrón de Salida
0	1000000000
1	0100000000
2	0010000000
3	0001000000
4	0000100000
5	0000010000
6	0000001000
7	0000000100
8	0000000010
9	0000000001

Cada topología de red se entrenó con un conjunto de 1,170 muestras, lo cual se conoce como el conjunto de entrenamiento. El conjunto de entrenamiento consiste del 90% del conjunto total. Toda topología de red fue entrenada con una tasa de aprendizaje de 1.0 y por 10,000 épocas, una época siendo una iteración donde se le presentó el conjunto de entrenamiento entero.

En la etapa de pruebas (decodificación o reconocimiento), a cada topología de red se le hizo pruebas con 130 muestras, llamado el conjunto de pruebas, para encontrar el porcentaje de reconocimiento.

Las pruebas se hicieron con dos tipos de configuración de red. En la primera se hicieron las pruebas con arquitecturas donde se tenía solamente una capa oculta y variaban el número de neuronas en esta capa. La tabla 2 muestra los resultados con estas arquitecturas. La tabla muestra la configuración de topología, el porcentaje de reconocimiento de la red ya entrenada y el último error fue calculado con el error cuadrático medio (MSE) en la última época del entrenamiento. Se puede ver que las arquitecturas del 2 al 5 dieron resultados buenos, donde la quinta da el mayor resultado de porcentaje de reconocimiento. Se debe destacar que la arquitectura 5 dio el mejor porcentaje de reconocimiento de todas las pruebas con un 84.62%. La arquitectura 1 no convergió en algo y por tanto no reconoció ningún carácter.

Tabla 2 Topologías de Redes Neuronales de Una Capa Oculta.

Arquitectura	Topología	Porcentaje Reconocimiento	Ultimo Error MSE
1	35x70x10	0%	4.5
2	35x35x10	81.45%	0.008557006
3	35x17x10	81%	0.024637074
4	35x15x10	80.76%	0.034835098
5	35x12x10	84.62%	0.024740543

La figura 9 muestra cómo se disminuyó el error total de la red con forme se va entrenado. Como se puede ver en la figura no fue necesario entrenar la red por 10,000 épocas, las varias redes ya habían convergido antes de 4,000 épocas.

En la segunda configuración de red se hicieron las pruebas con arquitecturas donde se tenía dos capas ocultas y solo variaban el número de neuronas en estas, pero siempre con el mismo número en las dos capas. La tabla 3 muestra los resultados de estas. Como se puede ver en la tabla, ninguna topología dio buenos resultados. La arquitectura 3 obtuvo mejores resultados pero el 24% de

reconocimiento no es muy útil. La arquitectura 1 no convergió y no pudo reconocer algún carácter.



Figura 9 Resultados de Entrenamiento de Topologías de Una Capa Oculta.

Tabla 3 Topologías de Redes Neuronales de Dos Capas Ocultas.

Arquitectura	Topología	Porcentaje Reconocimiento	Ultimo Error MSE
1	35x35x35x10	0%	0.505441469
2	35x17x17x10	10%	0.15272143
3	35x15x15x10	24%	0.106048881
4	35x12x12x10	10%	0.27263727

La figura 10 muestra cómo fue cambiando el error total de cada red conforme fue avanzando el entrenamiento. La primera red, la de 35x35x35x10, fue disminuyendo su error, pero en las 10,000 épocas que tuvo no convergió. Las demás redes iniciaron con un error bajo y luego fue incrementando el error en vez de disminuir y tuvieron muchas oscilaciones



Figura 10 Resultados de Entrenamiento de Topologías de Dos Capas Ocultas.

6. Conclusiones

Se entrenó la red neuronal con varias topologías para encontrar la configuración más adecuada para el reconocimiento de dígitos manuscritos. La topología en la cual se tiene lo doble de neuronas de entrada en una sola capa oculta, 35x70x10, no converge en algo y por tanto no reconoció ninguno de los dígitos que se le presentaron en la etapa de calificación.

Las topologías de una sola capa oculta con neuronas entre 35 y 12 por lo regular dieron buenos porcentajes de reconocimiento. Pero se debe admitir que ninguna topología de red obtuvo un porcentaje de reconociendo mayor al 85%.

Las topologías de red con más de una capa oculta no dan buenos resultados, por lo regular inician con un error total pequeño y luego conforme va avanzando el entrenamiento el error incrementa en vez de disminuir y por tanto no convergen. Esto se debe en que en la fase de aprendizaje con el algoritmo de descenso de gradiente la red entra en un mínimo local del cual no se pueden salir a encontrar el mínimo global.

En un trabajo futuro se puede añadir una fase llamada pre-entrenamiento, esto con el fin de incrementar el porcentaje de reconocimiento. Esta fase de pre-entrenamiento utiliza algoritmos que previenen que una red entre en un mínimo local en su entrenamiento con el algoritmo de retro-propagación con descenso de gradiente.

En el reconocimiento de caracteres con redes neuronales existe una fase llamada extracción de características. En esta fase se utilizan algoritmos que extraen información importante de las imágenes, esto con el fin de mejorar el porcentaje de reconocimiento e incrementar la eficiencia de entrenamiento ya que se puede reducir el número de neuronas de entrada.

7. Bibliografía y Referencias

- [1] Y. Perwej, A. Chaturvedi, "Neural Networks for Handwritten English Alphabet Recognition". *International Journal of Computer Applications*. Vol. 20. No. 7. April 2014.

- [2] A. Pal, D. Singh, "Handwritten English Character Recognition Using Neural Networks," *International Journal of Computer Science & Communication*. Vol. 1. No. 2. July-December 2010.
- [3] C. Patel, "Handwritten Character Recognition Using Neural Networks". *International Journal of Scientific & Engineering Research*. Vol. 2. No. 5. May 2011.
- [4] J. J. Castaneyra-Matus, Reconocimiento de Dígitos del Código Postal Mexicano Mediante Redes Neuronales Artificiales. Tesis de Licenciatura. México: Universidad Tecnológica de la Mixteca.
- [5] G.R. Cónstenla, Reconocimiento Óptico de Dígitos con Redes Neuronales. Tesis de Doctorado, Argentina: Universidad del Belgrano. 2010.
- [6] C.J. Sánchez-Fernández, V.S. Consuegra, Reconocimiento Óptico de Caracteres (OCR). Madrid, España: Universidad Carlos III.
- [7] Y. Perwej, A. Chaturvedi, "Neural Networks for Handwritten English Alphabet Recognition". *International Journal of Computer Applications*. Vol. 20. No. 7. April 2014.
- [8] Umal Patel, "An Introduction to the Processo of Optical Character Recognition," *International Journal of Science and Research*. Vol. 2. No. 5. May 2013.
- [9] N. Casillas Gil, Sistema Basado En Redes Neuronales Para El Reconocimiento De Dígitos Manuscritos. Tesis de Licenciatura, Universidad Carlos III de Madrid. Madrid, España. 2012.
- [10] Tim Klassen, Towards Neural Network Recognition of Handwritten Arabic Letters. Halifax, Nova Scotia, Dalhousie University. 2001. Canada.
- [11] F. Bortolozzi, A. de Souza Britto, L. S. Oliveira, M. Morita, "Recent Advances in Hadwriting Recognition". 2006.
- [12] V. Sandu and Florin Leon, Recognition of Handwritten Digits Using Multilayer Perceptrons. *Buletinul Insitutului Politehnic Din Iasi*. 2009.
- [13] D. Kriesel, A Brief Introduction to Neural Networks. Germany: University of Bonn. 2005.

- [14] Christos Stergiou and Dimitrios Siganos. NEURAL NETWORKS. https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Contents.
- [15] Leticia María Seijas, Reconocimiento de patrones utilizando tecnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos. Tesis de Doctorado, Universidad de Buenos Aires. 2011. Buenos Aires, Argentina.
- [16] Weisstein, Eric W. "Linear Congruence Method." From MathWorld--A WolframWeb. <http://mathworld.wolfram.com/LinearCongruenceMethod.html>.
- [17] Handwriting Recognition, Jeff Heaton. <http://www.jeffheaton.com>. Febrero, 2016.
- [18] H. A. Legal-Ayala, J. Facon, "Image segmentation by learning approach". In Proceedings of the 7th International Conference on Document Analysis and Recognition. Vol.2. 2003. Pp 819.
- [19] Y. C. Jhing-FaWang, "Locating the destination address block on images of complex mail pieces". Journal of the Chinese Institute of Engineers. 2001.
- [20] J. J. Hull, S. N. Srihari, L. Kuan, P. Cullen, P. Palumbo, "A blackboard-based approach to handwritten Zip code recognition". In Proceedings of 3rd United States Postal Service Advanced Technology Conj. 1988. Pp. 1018-1032.

8. Autores

Juan Manuel Martínez Cabrera es alumno de la Licenciatura en Ingeniería de Software de la Universidad Autónoma de Zacatecas.

Dr. Aldonso Becerra Sánchez es Ingeniero en Sistemas Computacionales, tiene la Maestría en Ingeniería con orientación en Computación, y título de Doctor en Educación por parte de la Universidad Autónoma de Durango.

M.I.A. Santiago Esparza Guerrero es Ingeniero Eléctrico, tiene la Maestría en Informática Administrativa por parte de la Universidad Autónoma de Durango.

Dr. Gustavo Zepeda Valles es Ingeniero en Sistemas Computacionales, tiene la Maestría en Informática Administrativa, y título de Doctor en Educación por parte de la Universidad Autónoma de Durango.

M.C Uriel Ramírez-García Correa es Licenciado en sistemas Computacionales y Administrativos, tiene la Maestría en Ciencias por parte del Tecnológico de Monterrey.