

# REDES NEURONALES Y SU APLICACIÓN EN LA CLASIFICACIÓN DE PATRONES<sup>a</sup>

## NEURAL NETWORKS AND THEIR APPLICATION IN THE CLASSIFICATION OF PATTERNS

EDGAR ALAMILLA-JIMÉNEZ<sup>b</sup>, ADDY BOLÍVAR-CIMÉ<sup>b\*</sup>, EDILBERTO NÁJERA<sup>b</sup>

Recibido 23-10-2021, aceptado 24-12-2021, versión final 30-12-2021

Artículo Investigación

**RESUMEN:** El objetivo de este trabajo es mostrar algunas aplicaciones de redes neuronales en la clasificación de patrones. Se explicará la estructura de una red neuronal artificial, el gradiente descendente y el algoritmo de backpropagation. Se exhibirá cómo se ajustan redes neuronales a datos encontrados en la literatura, haciendo uso de la función sigmoide logístico y la función sigmoide arctangente, considerando capas ocultas y variando el número de neuronas ocultas; además se muestran tablas de confusión para comparar la clasificación de las redes neuronales.

**PALABRAS CLAVE:** Redes neuronales; gradiente descendente; algoritmo de backpropagation; reconocimiento de dígitos; reconocimiento de caras.

**ABSTRACT:** The objective of this work is to show some applications of neural networks for classifying patterns. It will be explain the structure of a neural network, gradient descent and backpropagation algorithm. It will be exhibited how neural networks are adjusted to real data found in the literature, taking into account the logistic sigmoid function and the arctangent sigmoid function, considering hidden layers and varying the number of hidden neurons; furthermore, confusion tables are shown to compare the classification of neural networks.

**KEYWORDS:** Neural network; gradient descent, backpropagation algorithm; digit recognition; faces recognition.

## 1. INTRODUCCIÓN

La ciencia de datos es hoy en día una herramienta fundamental para la explotación de datos y la generación de conocimiento. Entre los objetivos que persigue se encuentra la búsqueda de modelos que describan patrones y comportamientos a partir de los datos con el fin de tomar decisiones o hacer predicciones. Es un área que ha experimentado un enorme crecimiento al extenderse el acceso a grandes volúmenes de datos e incluso su tratamiento en tiempo real. Abarca a numerosos grupos de investigación de diferentes áreas (computación, estadística, matemáticas, ingeniería, etc.) que trabajan en la propuesta de nuevos algoritmos,

<sup>a</sup>E. Alamilla-Jiménez, E., Bolívar-Cimé, A. & Nájera, E.(2022). Redes neuronales y su aplicación en la clasificación de patrones. *Rev. Fac. Cienc.*, 11 (1), 78–97. DOI: <https://doi.org/10.15446/rev.fac.cienc.v11n1.99173>

<sup>b</sup>División Académica de Ciencias Básicas. Universidad Juárez Autónoma de Tabasco. Cunduacán, Tabasco, México.

\* Autor para la correspondencia: addy.bolivar@ujat.mx

técnicas de computación e infraestructuras para la captura, almacenamiento y procesado de datos, etc., ver García-Molina *et al.* (2018).

Una parte fundamental de la ciencia de datos son las redes neuronales artificiales, las cuales son modelos computacionales inspirados en los trabajos de investigación, iniciados en 1930, cuyo propósito era modelar computacionalmente el aprendizaje humano llevado a cabo a través de las neuronas en el cerebro. Las redes neuronales son una nueva forma de analizar la información de tal modo que son capaces de detectar y aprender patrones complejos y características de los datos. Como nuestro cerebro, aprenden de la experiencia y del pasado, y aplican el conocimiento adquirido para resolver problemas nuevos. Este aprendizaje se obtiene como resultado del entrenamiento (*training*), el cual permite la sencillez y la potencia de adaptación y evolución ante una realidad cambiante y muy dinámica. Una vez adiestradas, las redes neuronales pueden hacer previsiones, clasificaciones y segmentación. Además, en la mayoría de los casos, presentan una eficiencia y confiabilidad por lo menos similares a los métodos estadísticos y sistemas expertos. En los casos de muy alta complejidad, las redes neuronales son especialmente útiles dada la dificultad para modelar que tienen otras técnicas. Rosenblatt (1958) propuso el primer modelo precursor de redes neuronales, el perceptrón. Sin embargo, éste tenía capacidades muy limitadas, lo que trajo como consecuencia que, en la década de 1970, esta área de investigación fuera casi abandonada; no fue sino hasta la década de 1980, con el uso de hardware computacional, que se dio un auge en la investigación de redes neuronales, el cual persiste hasta el día de hoy, ver Rodríguez-Flores (2010).

En el problema de clasificación se tienen datos  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , donde  $x_i \in \mathbb{R}^d$ ,  $y_i \in \{1, 2, \dots, k\}$  y  $k$  representa el número de clases. Estos datos son llamados datos de entrenamiento. Se supone que los datos provienen de una distribución desconocida. El problema que se quiere resolver es el de clasificar un nuevo dato  $x$  en alguna de estas clases. Para ello se desea construir una regla de clasificación, la cual está dada por una función

$$g : \mathbb{R}^d \rightarrow \{1, 2, \dots, k\}$$

$$x \mapsto g(x).$$

En muchas situaciones es deseable realizar la clasificación de patrones, es decir, el reconocimiento de caracteres (escritos a mano o a máquina), de voz, de caras, de objetos, de huellas, entre otros.

En este artículo se presenta la forma en que se utilizan las redes neuronales artificiales en la clasificación de patrones, explicando brevemente su implementación y proporcionando algunos ejemplos de su aplicación. A continuación se describe como está dividido este trabajo. En la sección 2 se presentan el perceptrón multicapa, el gradiente descendente y el algoritmo de *backpropagation* (propagación hacia atrás), que son piezas claves para la implementación de las redes neuronales artificiales. En la sección 3 se muestran ejemplos de aplicación con datos reales encontrados en la literatura, para ilustrar el uso de las redes neuronales artificiales como una herramienta para clasificar patrones, para lo cual se empleó la paquetería

“*nnet*”, ver Ripley & Venable, W. (2021), del software RStudio. Por último se presenta una sección de conclusiones.

## 2. REDES NEURONALES

### 2.1. Perceptrón multicapa

Como se menciona en Devroye *et al.* (2013), el discriminante lineal o perceptrón tiene la regla de decisión

$$\phi(x) = \begin{cases} 0, & \text{si } \psi(x) \leq \frac{1}{2}, \\ 1, & \text{en otro caso,} \end{cases}$$

basada en una combinación lineal  $\psi(x)$ , dada por:

$$\psi(x) = c_0 + \sum_{i=1}^d c_i x^{(i)} = c_0 + c^T x, \quad (1)$$

donde las  $c_i$ 's son los pesos,  $x = (x^{(1)}, \dots, x^{(d)})^T$  y  $c = (c_1, \dots, c_d)^T$ . Esto es llamado una red neuronal sin capas ocultas, ver Figura 1.

En una red neuronal (hacia adelante) o perceptrón multicapa con una capa oculta se tiene

$$\psi(x) = c_0 + \sum_{i=1}^k c_i \sigma(\psi_i(x)), \quad (2)$$

donde las  $c_i$ 's son como antes y cada  $\psi_i$  es de la forma dada en (1):  $\psi_i(x) = b_i + \sum_{j=1}^d a_{ij} x^{(j)}$  para algunas constantes  $b_i$  y  $a_{ij}$ , ver Figura 2. La función  $\sigma$  es llamada sigmoide, la cual se define como una función no decreciente con  $\sigma \rightarrow -1$  cuando  $x \downarrow -\infty$  y  $\sigma \rightarrow 1$  cuando  $x \uparrow \infty$ . Algunos ejemplos de sigmoides son: el sigmode umbral, el sigmoide estándar o logístico, el sigmoide arctangente y el sigmoide gaussiano. En el

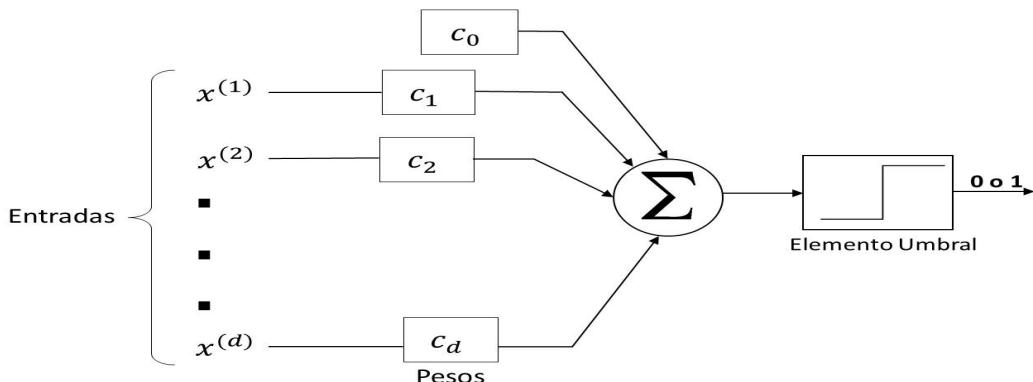


Figura 1: El perceptrón de Rosenblatt. La decisión se basa en una combinación lineal de los componentes del vector de entrada.

Fuente: Elaboración propia.

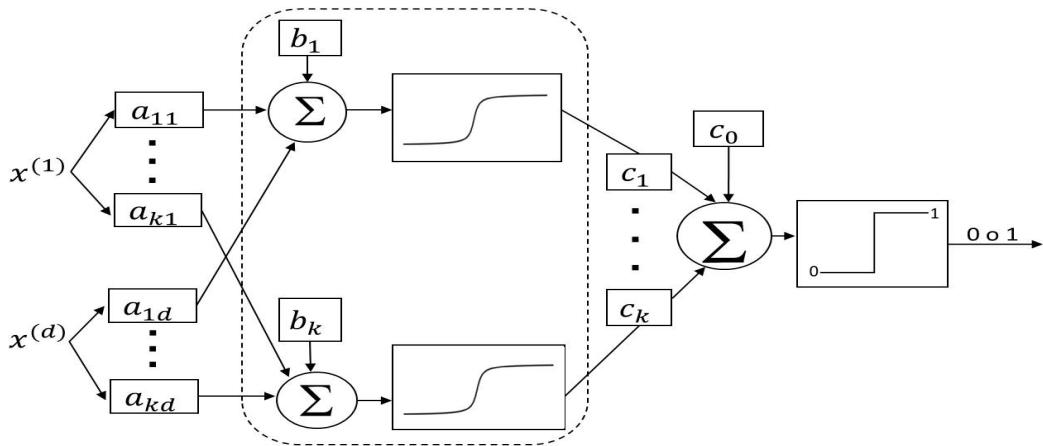


Figura 2: Red neuronal con una capa oculta. Las neuronas ocultas se encuentran dentro del marco. Fuente: Elaboración propia.

perceptrón con una capa oculta, se dice que hay  $k$  neuronas ocultas (la salida de la  $i$ -ésima neurona oculta es  $u_i = \sigma(\psi_i(x))$ ). Así, la expresión (2) puede reescribirse como

$$\psi(x) = c_0 + \sum_{i=1}^k c_i u_i,$$

que es similar a la expresión (1). Se puede continuar este proceso y crear redes neuronales multicapa hacia adelante. Por ejemplo, un perceptrón de dos capas ocultas es de la forma:

$$\psi(x) = c_0 + \sum_{i=1}^{\ell} c_i z_i,$$

donde

$$z_i = \sigma \left( d_{i0} + \sum_{j=1}^k d_{ij} u_j \right), \quad 1 \leq i \leq \ell,$$

y

$$u_j = \sigma \left( b_j + \sum_{i=1}^d a_{ji} x^{(i)} \right), \quad 1 \leq j \leq k,$$

y las  $d_{ij}$ 's,  $b_j$ 's y  $a_{ji}$ 's son constantes. La primera capa oculta tiene  $k$  neuronas ocultas, mientras que la segunda capa oculta tiene  $\ell$  neuronas ocultas, ver Figura 3.

## 2.2. Gradiente descendente

Como se menciona en Alpaydin (2014), en la clasificación basada en verosimilitud, los estimadores son estadísticos suficientes de la función de densidad de probabilidad dada la clase,  $p(x|C^{(i)})$ , y de la probabilidad a priori de la clase,  $P(C^{(i)})$ ; el método que se utiliza para estimar los parámetros es el de máxima verosimilitud. Mientras que en discriminación, los parámetros son de los discriminantes y están optimizados para

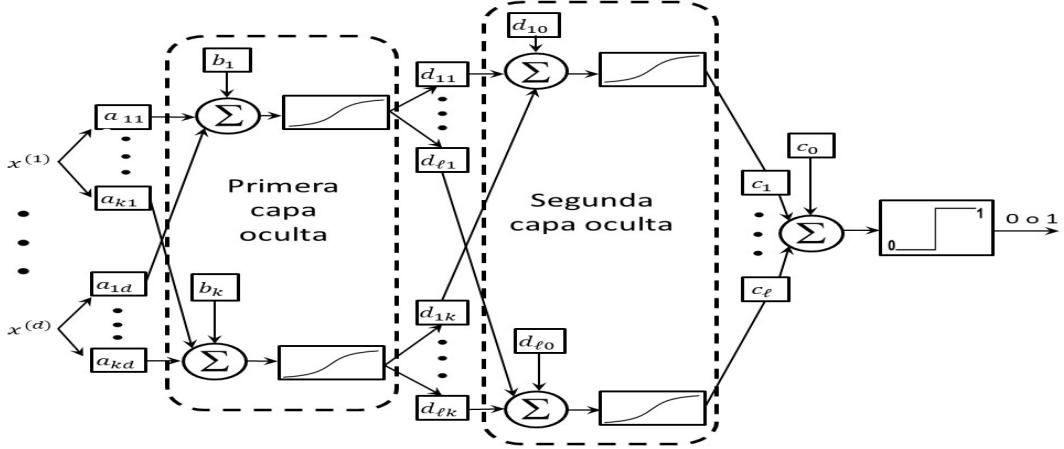


Figura 3: Red neuronal hacia adelante con dos capa ocultas. Fuente: Elaboración propia.

minimizar el error de clasificación.

Cuando  $w = (w^{(1)}, \dots, w^{(d)})^T$  denota el vector de pesos o el conjunto de parámetros y  $\mathbf{E}(w|X)$  es el error con parámetros  $w$  para el conjunto de entrenamiento  $X$  dado, se busca:

$$w^* = \arg \min_w \mathbf{E}(w|X).$$

En muchos casos no existe una solución analítica y se debe recurrir a métodos de optimización iterativos, de los cuales el método más empleado es el gradiente descendente. Cuando  $\mathbf{E}(w)$  es una función diferenciable de un vector de variables, se tiene el vector gradiente, compuesto de las derivadas parciales

$$\nabla_w \mathbf{E} = \left[ \frac{\partial \mathbf{E}}{\partial w_1}, \frac{\partial \mathbf{E}}{\partial w_2}, \dots, \frac{\partial \mathbf{E}}{\partial w_d} \right]^T.$$

El procedimiento del gradiente descendente para minimizar  $\mathbf{E}$  comienza a partir de un  $w$  aleatorio y en cada paso actualiza  $w$  en la dirección opuesta del gradiente:

$$\Delta w_i = -\eta \frac{\partial \mathbf{E}}{\partial w_i}, \quad \forall i, \quad (3)$$

por lo que

$$w_i = w_i + \Delta w_i, \quad (4)$$

donde  $\eta$  es llamado el tamaño de paso o factor de aprendizaje y determina cuánto moverse en esa dirección. Cuando se llega a un mínimo, la derivada es 0 y el procedimiento termina. Esto indica que el procedimiento encuentra el mínimo más cercano que puede ser un mínimo local, y no hay garantía de encontrar el mínimo global, a menos que la función tenga solo un mínimo. El uso de un buen valor para  $\eta$  también es crítico; si es demasiado pequeño la convergencia puede ser demasiado lenta, pero un valor grande puede causar oscilaciones e incluso divergencia.

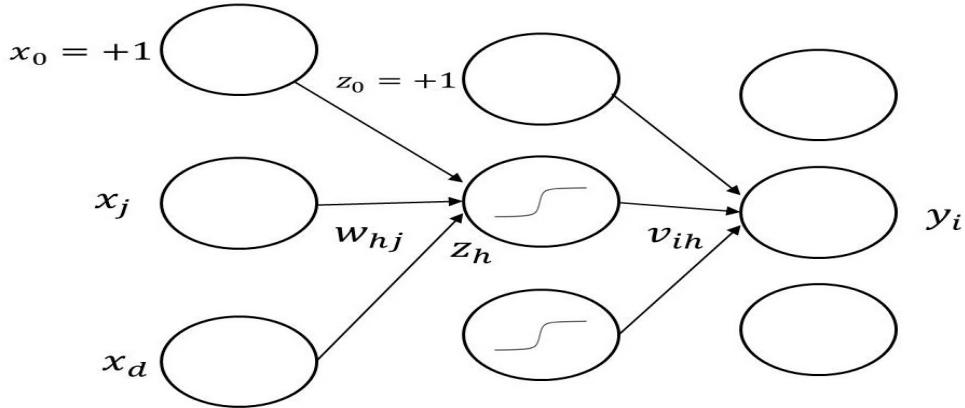


Figura 4: Estructura de un perceptrón multicapa. Fuente: Elaboración propia.

### 2.3. El algoritmo de *backpropagation*

El entrenamiento de un perceptrón multicapa es análogo al de un perceptrón, la única diferencia es que ahora la salida es una función no lineal de las entradas debido a la función sigmoide en las neuronas ocultas.

En la Figura 4 se muestra la estructura de un perceptrón multicapa en la cual las  $x_j$ ,  $j = 0, \dots, d$ , son las entradas y las  $z_h$ ,  $h = 1, \dots, H$ , son las neuronas ocultas, donde  $H$  es la dimensión de este espacio oculto,  $z_0$  es el intercepto de la capa oculta, las  $y_i$ ,  $i = 1, \dots, k$ , son las unidades de salida, las  $w_{hj}$  son los pesos de la primera capa, y las  $v_{ih}$  son los pesos de la segunda capa. Como se observa en esta figura, se consideran las neuronas ocultas como entradas de la segunda capa la cual es un perceptrón, que se ajusta de la forma usual dadas las entradas  $z_h$ . Para los pesos de la primera capa,  $w_{hj}$ , se utiliza la regla de la cadena para calcular la derivada parcial

$$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{hj}}.$$

Es como si el error se propagara desde la salida y de regreso a las entradas, y por esa razón se le denominó backpropagation, ver Alpaydin (2014). Con la fórmula anterior se calculan las entradas del vector gradiente y se lleva a cabo el método del gradiente descendente.

#### 2.3.1. Discriminación de dos clases

Cuando hay dos clases, una unidad de salida es suficiente y puede ser vista de la siguiente forma:

$$y^t = \sigma \left( \sum_{h=1}^H v_h z_h^t + v_0 \right),$$

que aproxima a  $P(C^{(1)}|x^t)$ , la probabilidad de que el dato  $x^t$  pertenezca a la clase 1, y  $\hat{P}(C^{(2)}|x^t) \equiv 1 - y^t$  es la probabilidad de que el dato  $x^t$  pertenezca a la clase 2. La función de error, mejor conocida como entropía cruzada, es

$$E(W, v|X) = - \sum_t r^t \log y^t + (1 - r^t) \log(1 - y^t),$$

donde  $W$  es la matriz de pesos en las capas ocultas,  $v$  es el vector de pesos en la capa de salida (ver Figura 4) y  $r^t$  es la etiqueta correspondiente al dato  $x^t$ .

Las ecuaciones de actualización implementando el gradiente descendente son:

$$\Delta v_h = \eta \sum_t (r^t - y^t) z_h^t$$

y

$$\Delta w_{hj} = \eta \sum_t (r^t - y^t) v_{ih} z_h^t (1 - z_h^t) x_j^t.$$

### 2.3.2. Discriminación multiclas

En un problema de clasificación multiclas ( $k > 2$ ) hay  $k$  salidas. Se define

$$o_i^t = \sum_{h=1}^H v_{ih} z_h^t + v_{i0}, \quad i = 1, 2, \dots, k,$$

y se utiliza la función *softmax* para indicar la dependencia entre clases,

$$y_i^t = \frac{\exp o_i^t}{\sum_j \exp o_j^t}, \quad i = 1, 2, \dots, k,$$

donde  $y_i^t$  aproxima a  $P(C^{(i)}|x^t)$ . La función de error es

$$\mathbf{E}(W, V|X) = - \sum_t \sum_i r_i^t \log y_i^t,$$

donde  $V$  es la matriz de los pesos en la capa de las salidas, de aquí se obtienen las ecuaciones de actualización utilizando el gradiente descendente:

$$\Delta v_{ih} = \eta \sum_t (r_i^t - y_i^t) z_h^t$$

y

$$\Delta w_{hj} = \eta \sum_t \left[ \sum_i (r_i^t - y_i^t) v_{ih} \right] z_h^t (1 - z_h^t) x_j^t.$$

### 2.3.3. Múltiples capas ocultas

Como se ha mencionado anteriormente, es posible tener múltiples capas ocultas, cada una con sus propios pesos y en la que se aplica la función sigmoide a su suma ponderada. Para clasificación, por ejemplo de dos clases, si se tiene un perceptrón multicapa con dos capas ocultas, se escribe:

$$z_{1h} = \sigma(w_{1h}^T x) = \sigma \left( \sum_{j=1}^d w_{1hj} x_j + w_{1h0} \right), \quad h = 1, \dots, H_1,$$

$$z_{2\ell} = \sigma(w_{2\ell}^T z_{1h}) = \sigma \left( \sum_{h=0}^{H_1} w_{2\ell h} z_{1h} + w_{2\ell 0} \right), \quad \ell = 1, \dots, H_2,$$

$$y = \sigma \left( \sum_{\ell=1}^{H_2} v_{\ell} z_{2\ell} + v_0 \right),$$

donde  $w_{1h}$  y  $w_{2\ell}$  son los pesos de la primera y segunda capa,  $z_{1h}$  y  $z_{2\ell}$  son las neuronas de la primera y segunda capa oculta, y  $v$  es el vector cuyas entradas son los pesos de la tercera capa. En una red de este tipo, para entrenar los pesos de la primera capa se necesita propagar hacia atrás (*backpropagate*) una capa más.

### 3. RESULTADOS

En esta sección se mostrarán algunas aplicaciones de redes neuronales en la clasificación de patrones con datos encontrados en la literatura. Estos datos son divididos en datos de entrenamiento y de prueba. Con los datos de entrenamiento se ajustan redes neuronales para clasificación utilizando la función sigmoide logística y la función sigmoide arcotangente. Posteriormente, se utilizan los datos de prueba para evaluar el modelo de la red neuronal para la clasificación de patrones a través de tablas de confusión, y se comparan las proporciones de error de clasificación correspondientes a las dos funciones sigmoide en estudio.

#### 3.1. Clasificación de dígitos escritos a mano

Se cuenta con datos de dígitos escritos manualmente tomados del conjunto de datos del Modified National Institute of Standards and Technology (MNIST), ver Bishop (2006), que a su vez se construyó modificando un subconjunto de un conjunto de datos mucho más grande producido por NIST (the National Institute of Standards and Technology). La base de datos consiste de un conjunto de entrenamiento de 60,000 muestras y un conjunto de prueba de 10,000 muestras. Algunos datos fueron recopilados de empleados de la oficina del censo y el resto fueron recopilados de niños de secundaria. Se tuvo cuidado de garantizar que las muestra del conjunto de prueba fueran escritos por personas diferentes a las de las muestras del conjunto de entrenamiento.

Los datos originales del NIST tenían píxeles binarios (blanco o negro). Para crear los datos del MNIST, se normalizó el tamaño de estas imágenes para que quepan en una imagen de  $20 \times 20$  píxeles, mientras se conserva su relación de aspecto. En consecuencia, debido al suavizado (anti-aliasing) utilizado para cambiar su resolución, las imágenes de los dígitos del MNIST resultaron tener escalas de grises. Luego, estas imágenes se centraron en una imagen de  $28 \times 28$  píxeles. Algunos ejemplos se muestran en la Figura 5.

Cada imagen es transformada en un vector de entradas reales de dimensión  $784 = 28 \times 28$ , mediante la concatenación de columnas. En la Figura 6 se puede apreciar un ejemplo de ello.

La motivación de este ejemplo es mostrar que la clasificación de dígitos puede ser automatizada mediante redes neuronales artificiales de manera adecuada. El proceso de clasificar grandes cantidades de dígitos

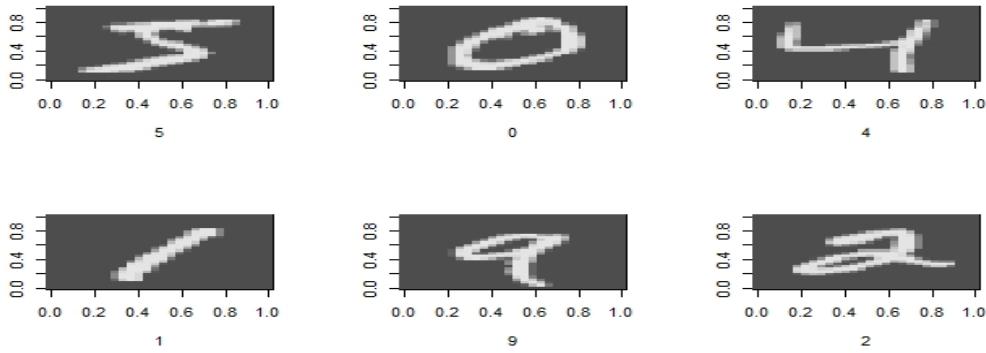


Figura 5: Imágenes de escritura de dígitos del MNIST de  $28 \times 28$  píxeles en escalas de grises. Fuente: Elaboración propia.

se puede efectuar en lapsos de tiempo relativamente cortos, además el procesamiento es realizado por un equipo de cómputo y el trabajo de catalogar los dígitos se realiza de manera objetiva. En cambio, un grupo de personas clasificaría una cantidad grande de dígitos en un tiempo extenso, además de que podrían haber errores debido al agotamiento o por carencia de objetividad en dicho proceso.

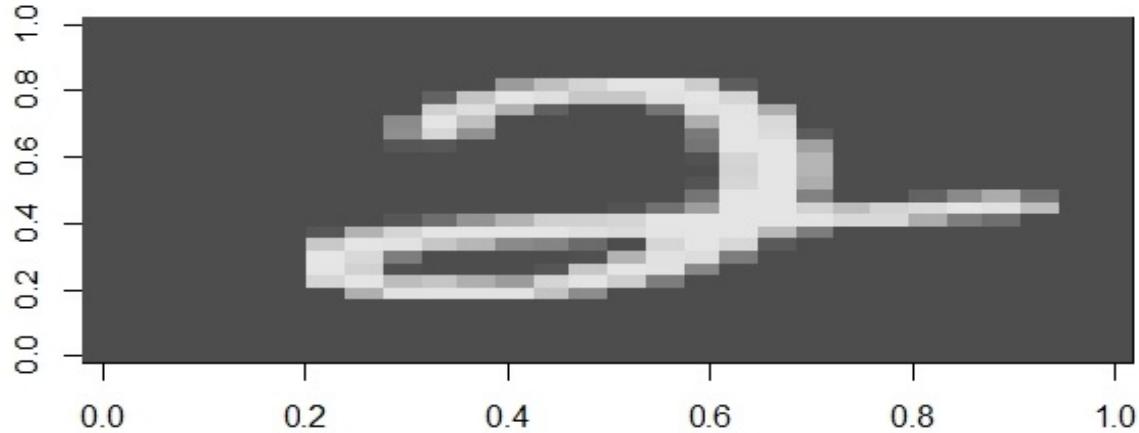
Como se menciona en Hastie *et al.* (2009), la clasificación de dígitos escritos manualmente se puede utilizar en la implementación de un proceso de ordenamiento automático de sobres de acuerdo a su código postal. Otras aplicaciones son el reconocimiento de números de placa y en el procesamiento de cheques bancarios, ver Dixit *et al.* (2020).

Utilizando los datos de entrenamiento se ajustaron redes neuronales usando la función sigmoide logística y la función sigmoide arcotangente, mediante la metodología descrita en la Sección 2. Cabe mencionar que se consideró un tamaño de paso de  $\eta = 0.001$  en el gradiente descendente, las entradas del vector de pesos inicial  $w$  se eligieron aleatoriamente en el intervalo  $[-0.5, 0.5]$  y se tomó como número máximo de iteraciones a 100,000. En cada caso se consideró una capa oculta y el número de neuronas ocultas se varió en 3, 6, 10, 13, 15 y 20. Con los datos de pruebas y el ajuste de la red neuronal, se obtuvieron tablas de confusión. En las Tablas, 1–6 se presentan las tablas de confusión correspondientes a 3, 10 y 20 neuronas ocultas. En estas tablas “Reales” se refiere a los verdaderos valores de los dígitos; “Predichos” a los valores predichos por la red neuronal; los valores dentro de la tabla proporcionan las frecuencias en la clasificación.

Después de obtener las tablas de confusión se obtuvieron las proporciones de error de clasificación, las cuales se proporcionan en la Tabla 7.

Se observa en las tablas de confusión con tres neuronas ocultas, y cualquiera de las funciones sigmoide, que la red neuronal no clasifica los dígitos en todas las clases (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), únicamente en unas

(a) Vista concatenada



(b) Vista en escalas de grises

Figura 6: Visualización del dígito 2. Fuente: Elaboración propia.

cuantas. A partir de seis neuronas ocultas ya se observan todas las clases. En la tabla de proporciones de error de clasificación se nota que no es recomendable considerar tres o seis neuronas ocultas, debido a que con ambas funciones sigmoide las proporciones de error tienen un valor muy elevado. Sin embargo, para esta clase de datos, a partir de diez neuronas ocultas no hay diferencia significativa entre las dos funciones sigmoide, porque las tablas de confusión resultan ser similares o existe una pequeña diferencia entre ellas, además de tener proporciones de error significativamente pequeñas, por lo que puede utilizarse cualquiera de ellas para ajustar la red neuronal que produzca buenos resultados. De igual forma, si se desea tener proporciones de error más pequeñas se tendrían que aplicar más neuronas ocultas dentro de la capa oculta, por lo que esto se deja a consideración del experimentador.

Tabla 1: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 3 neuronas ocultas.

		Predichos						
		1	2	3	4	6	7	
Reales		0	1	59	910	2	1	7
0		1	59	910	2	1	7	
1		1077	4	39	1	13	1	
2		16	699	179	28	105	5	
3		7	54	928	3	4	14	
4		8	13	2	822	25	112	
5		6	76	787	1	5	17	
6		8	95	21	36	798	0	
7		19	3	55	19	5	927	
8		18	64	832	11	11	38	
9		5	4	43	57	2	898	

Tabla 3: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 10 neuronas ocultas.

		Predichos									
		0	1	2	3	4	5	6	7	8	9
Reales		0	1	5	3	1	3	6	5	9	1
0		947	0	5	3	1	3	6	5	9	1
1		0	1096	4	6	0	3	3	3	17	3
2		19	16	888	30	14	2	24	11	20	8
3		12	10	31	832	5	42	2	19	47	10
4		2	1	6	1	905	2	16	4	3	42
5		20	8	5	40	17	708	15	11	57	11
6		15	1	12	1	9	12	896	3	9	0
7		5	16	34	4	14	1	0	923	12	19
8		8	13	15	23	19	25	33	7	809	22
9		6	5	4	7	56	3	7	38	28	855

Tabla 5: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 20 neuronas ocultas.

		Predichos									
		0	1	2	3	4	5	6	7	8	9
Reales		0	1	6	3	7	7	8	1	4	3
0		941	0	6	3	7	7	8	1	4	3
1		0	1111	3	3	0	1	5	2	10	0
2		18	6	920	10	15	5	17	13	25	3
3		4	0	21	912	0	25	7	15	21	5
4		4	4	4	4	875	3	16	6	8	58
5		12	3	5	28	10	784	13	7	23	7
6		9	6	15	3	14	14	891	0	5	1
7		4	5	20	9	4	4	0	949	8	25
8		6	1	16	30	14	16	11	9	859	12
9		8	4	2	15	58	5	2	14	11	890

Finalmente, se muestran ejemplos de cómo la red neuronal clasifica dígitos, los cuales se presentan en las Figuras 7 y 8.

Tabla 2: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 3 neuronas ocultas.

		Predichos				
		0	1	4	7	
Reales		0	956	21	2	1
0		956	21	2	1	
1		3	1130	0	2	
2		764	228	29	11	
3		59	928	1	22	
4		36	20	678	248	
5		106	745	12	29	
6		926	17	14	1	
7		12	66	40	910	
8		70	873	2	29	
9		19	35	414	541	

Tabla 4: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 10 neuronas ocultas.

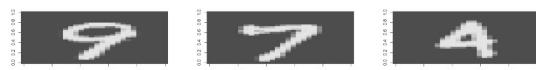
		Predichos									
		0	1	2	3	4	5	6	7	8	9
Reales		0	1	2	3	4	5	6	7	8	5
0		928	0	4	3	1	17	18	2	2	5
1		0	1074	8	16	0	3	2	7	23	2
2		25	13	889	13	11	10	18	22	29	2
3		2	4	53	839	1	57	1	18	24	11
4		3	4	3	3	865	6	9	7	15	67
5		9	1	11	56	13	692	18	14	56	22
6		14	0	18	2	17	31	867	0	6	3
7		3	23	15	14	10	8	2	897	10	46
8		9	15	15	29	11	36	29	15	752	63
9		7	3	0	10	37	20	3	24	17	888

Tabla 6: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 20 neuronas ocultas.

		Predichos										
		0	1	2	3	4	5	6	7	8	9	
Reales		0	939	1	1	2	2	12	11	8	4	0
0		939	1	1	2	2	12	11	8	4	0	
1		0	1107	5	3	0	3	5	3	8	1	
2		8	9	910	31	17	5	11	14	25	2	
3		3	3	25	914	1	14	5	14	24	7	
4		1	0	8	0	923	1	21	2	2	24	
5		6	2	3	59	3	764	20	8	20	7	
6		17	6	8	1	12	22	882	1	9	0	
7		0	6	16	16	10	2	0	928	3	47	
8		7	8	4	24	15	16	16	12	851	21	
9		9	5	1	8	43	7	2	27	17	890	

Tabla 7: Proporciones de error de clasificación considerando una capa oculta y variando en 3, 6, 10, 13, 15 y 20 el número de neuronas ocultas.

Neuronas ocultas	Función sigmoide	Logística	Arcotangente
3		0.4749	0.6326
6		0.2065	0.1704
10		0.1141	0.1309
13		0.1139	0.1046
15		0.0899	0.1028
20		0.0868	0.0892

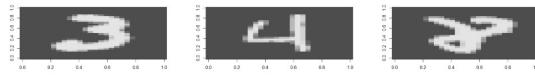


(a) Dígito 9, predicción 9. (b) Dígito 7, predicción 7. (c) Dígito 4, predicción 9.

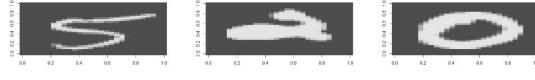


(d) Dígito 8, predicción 8. (e) Dígito 6, predicción 6. (f) Dígito 1, predicción 1.

Figura 7: Clasificación de dígitos correspondiente a una red neuronal de 1 capa oculta con 20 neuronas ocultas y función sigmoide logística. Fuente: Elaboración propia.



(a) Dígito 3, predicción 3. (b) Dígito 4, predicción 4. (c) Dígito 8, predicción 8.



(d) Dígito 5, predicción 5. (e) Dígito 2, predicción 4. (f) Dígito 0, predicción 0.

Figura 8: Clasificación de dígitos correspondiente a una red neuronal de 1 capa oculta con 20 neuronas ocultas y función sigmoide arcotangente. Fuente: Elaboración propia.

### 3.2. Reconocimiento de caras

Se cuenta con datos de imágenes de rostros de varias personas en varias poses, los cuales son proporcionados por Mitchell (1997). La base de datos consiste de imágenes de 20 personas diferentes, con aproximadamente 32 imágenes por persona variando su estado de ánimo (feliz, triste, enojado, neutral), la dirección en la que miran (izquierda, derecha, de frente, arriba) y si portan o no gafas de sol, tal como puede verse en la Figura 9. En total se recopilaron 624 imágenes en escala de grises, cada una con una resolución de  $64 \times 60$ , con cada píxel de imagen descrito por un valor de la intensidad en la escala de grises entre 0 (negro) y 255 (blanco). Cada imagen fue transformada en un vector de entradas reales de dimensión  $3840 = 64 \times 60$ , mediante la concatenación de columnas.

El propósito de este ejemplo es mostrar que la clasificación de sujetos puede ser automatizada mediante redes neuronales artificiales. El proceso de clasificar grandes cantidades de caras se puede efectuar en lapsos de tiempo relativamente cortos; el procesamiento es realizado por un equipo de cómputo y el trabajo de catalogar las caras se realiza de manera objetiva. De igual forma, la misma base de datos puede ser utilizada para clasificar a los sujetos de acuerdo a si portan o no gafas sol en la imagen. Del mismo modo, los datos

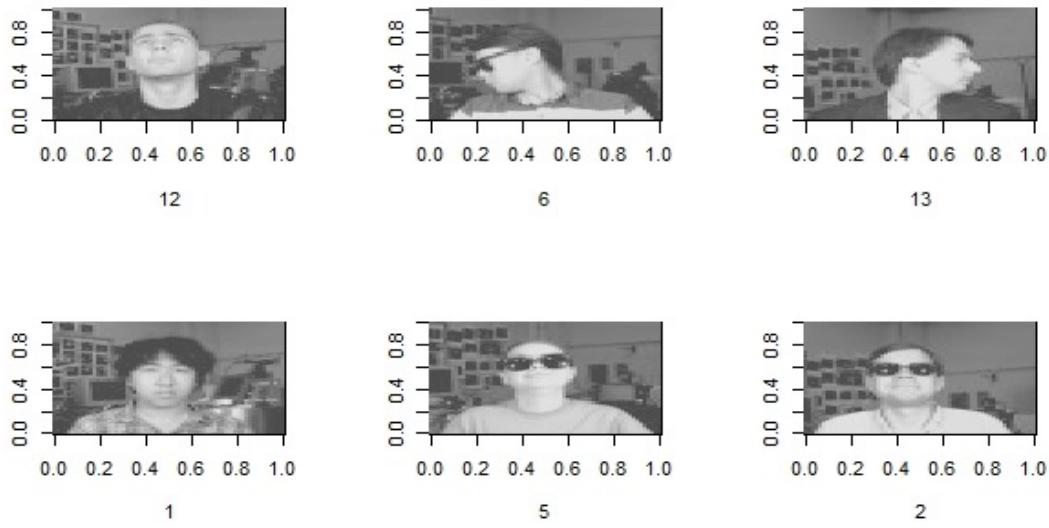


Figura 9: Imágenes de rostros de varias personas de  $64 \times 60$  píxeles en diferentes poses, diferentes estados de ánimos y si portan o no gafas de sol. Fuente: Elaboración propia.

son útiles para clasificar la dirección en la que miran los sujetos, es decir, izquierda, derecha, de frente o arriba. Así mismo, la base de datos puede ser empleada para clasificar el estado de ánimo de los sujetos, es decir, enojado, feliz, neutral o triste.

Cabe mencionar que la clasificación de caras es útil en procedimientos de seguridad, videovigilancia, interacción persona-computadora con fines de entretenimiento, robótica y tarjetas “inteligentes” (pasaportes, licencias de conducir, registro de votantes), ver Izenman (2013).

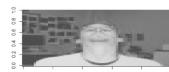
En cada análisis que se presenta se procedió a dividir de forma aleatoria en dos partes a la base de datos, el 70% de los puntos como datos de entrenamiento (436 puntos) y el 30% restante (188 puntos) conformaron los datos de prueba. Utilizando los datos de entrenamiento se ajustaron redes neuronales utilizando la función sigmoide logística y la función sigmoide arcotangente, mediante la metodología descrita en la Sección 2. Como anteriormente, se consideró un tamaño de paso de  $\eta = 0.001$  en el gradiente descendente, las entradas del vector de pesos inicial  $w$  se eligieron aleatoriamente en el intervalo  $[-0.5, 0.5]$  y se tomó como número máximo de iteraciones a 100,000. Con los datos de prueba y el ajuste de la red neuronal, se obtuvieron tablas de confusión para evaluar la clasificación proporcionada por las redes neuronales ajustadas.

### 3.2.1. Clasificación de sujetos

En este ejemplo se procedió a reconocer a un sujeto en particular de entre los demás, para ello se etiquetó al sujeto de interés, en este caso al sujeto “an2i”, con la etiqueta 1 y a los demás sujetos que conforman la

Tabla 8: Tabla de confusión de la red neuronal con función sigmoide logística y arcotangente de 1 capa oculta con 1 neurona oculta.

		Predichos	
		1	2
Reales	1	11	0
	2	0	177



(a) Etiqueta 2, predicho 2. (b) Etiqueta 2, predicho 2. (c) Etiqueta 1, predicho 1.



(b) Etiqueta 2, predicho 2. (c) Etiqueta 1, predicho 1.



(c) Etiqueta 1, predicho 1.



(d) Etiqueta 2, predicho 2. (e) Etiqueta 1, predicho 1. (f) Etiqueta 2, predicho 2.



(e) Etiqueta 1, predicho 1. (f) Etiqueta 2, predicho 2.



(f) Etiqueta 2, predicho 2.

Figura 10: Clasificación de sujetos correspondiente a una red neuronal de 1 capa oculta con 1 neurona oculta y función sigmoide logística. “sujeto an2i”=1, “otro”=2.

Fuente: Elaboración propia.

base de datos con la etiqueta 2.

Se ajustaron redes neuronales utilizando la función sigmoide logística y la función sigmoide arcotangente considerando una capa oculta con una neurona oculta. Las tablas de confusión fueron idénticas al considerar las dos funciones sigmoide, en la Tabla 8 se presenta la tabla de confusión obtenida.

Mediante la tabla de confusión se observa que la proporción de error con ambas funciones sigmoide es 0. Entonces para estos datos es suficiente considerar una red neuronal de una capa oculta, con una neurona oculta y cualquiera de las funciones sigmoide.

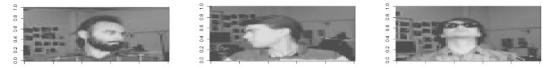
En las Figuras 10 y 11 se muestra cómo cada red neuronal clasifica a los sujetos.

### 3.2.2. Clasificación de uso de gafas de sol

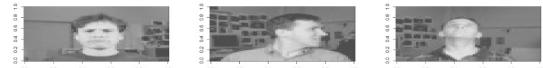
Para la clasificación de uso de gafas de sol de los sujetos, se procedió a etiquetarlos de la siguiente manera: “sin gafas de sol”=1, “con gafas de sol”=2. Se ajustaron redes neuronales utilizando la función sigmoide logística y la función sigmoide arcotangente; en cada caso se consideró una capa oculta con una neurona oculta. Las tablas de confusión fueron idénticas al considerar las dos funciones sigmoide; en la Tabla 9 se presenta la tabla de confusión obtenida.

Tabla 9: Tabla de confusión de la red neuronal con función sigmoide logística y arcotangente de 1 capa oculta con 1 neurona oculta.

		Predichos	
		1	2
Reales	1	91	1
	2	1	95



(a) Gafas 1, predicho 2. (b) Gafas 1, predicho 1. (c) Gafas 2, predicho 2.



(d) Gafas 1, predicho 1. (e) Gafas 1, predicho 1. (f) Gafas 1, predicho 1.



(a) Gafas 1, predicho 1. (b) Gafas 2, predicho 2. (c) Gafas 1, predicho 1.



(d) Gafas 2, predicho 2. (e) Gafas 1, predicho 1. (f) Gafas 2, predicho 1.

Figura 12: Clasificación de sujetos que portan gafas de sol o no, correspondiente a una red neuronal de 1 capa oculta con 1 neurona oculta y función sigmoide logística. “sin gafas de sol”=1, “con gafas de sol”=2. Fuente: Elaboración propia.

Figura 13: Clasificación de sujetos que portan gafas de sol o no, correspondiente a una red neuronal de 1 capa oculta con 1 neurona oculta y función sigmoide arcotangente. “sin gafas de sol”=1, “con gafas de sol”=2. Fuente: Elaboración propia.

Mediante la tabla de confusión se tiene que la proporción de error con ambas funciones sigmoide es 0.0106. Cabe mencionar que al aumentar el número de neuronas ocultas dentro de la capa oculta se obtuvo la misma proporción de error que con una neurona oculta, con ambas funciones sigmoide. Por lo que, considerando estos datos, es suficiente utilizar una red neuronal de una capa oculta, con una sola neurona oculta y cualquiera de las dos funciones sigmoide para obtener una buena clasificación de sujetos con gafas de sol.

Por último, se muestran ejemplos de cómo la red neuronal clasifica a los sujetos que portan gafas de sol o no en las imágenes, las cuales se presentan en las figuras 12 y 13.

### 3.2.3. Clasificación de dirección de miradas

Para el caso de clasificar la dirección en la que miran los sujetos, se etiquetó dicha dirección de la siguiente forma: “izquierda”=1, “derecha”=2, “de frente”=3 y “arriba”=4. Se ajustaron redes neuronales usando la función sigmoide logística y la función sigmoide arcotangente. En cada caso se consideró una capa oculta, y se varió el número de neuronas ocultas de 1 al 6.

En las tablas 10 y 11 se presentan las tablas de confusión para 1 neurona oculta, usando la función sigmoide

Tabla 10: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 1 neurona oculta.

		Predichos			
		1	2	3	4
Reales	1	41	0	0	0
	2	6	31	11	0
3	5	15	25	7	
4	0	0	0	47	

Tabla 12: Tabla de confusión de la red neuronal con función sigmoide logística y arcotangente de 1 capa oculta con 2 neuronas ocultas.

		Predichos			
		1	2	3	4
Reales	1	40	1	0	0
	2	0	48	0	0
3	1	0	49	2	
4	0	0	0	47	

Tabla 11: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 1 neurona oculta.

		Predichos			
		1	2	3	4
Reales	1	40	1	0	0
	2	0	48	0	0
3	3	6	30	13	
4	5	1	2	39	

Tabla 13: Tabla de confusión de la red neuronal con función sigmoide logística y arcotangente de 1 capa oculta con 3 neuronas ocultas.

		Predichos			
		1	2	3	4
Reales	1	40	1	0	0
	2	0	47	1	0
3	0	1	47	4	
4	0	0	0	47	

lógica y arcotangente, respectivamente. A partir de dos neuronas ocultas las tablas de confusión fueron idénticas al considerar las dos funciones sigmoide.

En las tablas 12 y 13 se presentan las tablas de confusión correspondientes a 2 y 3 neuronas ocultas, respectivamente.

Una vez obtenidas las tablas de confusión se obtuvieron las proporciones de error de clasificación que se muestran en la Tabla 14.

Tabla 14: Proporciones de error de clasificación considerando una capa oculta y variando de 1 a 6 neuronas ocultas.

		Función sigmoide		Logística	Arcotangente
		Neuronas ocultas			
1			0.2340	0.1649	
2			0.0213	0.0213	
3			0.0372	0.0372	
4			0.0372	0.0372	
5			0.0213	0.0213	
6			0.0372	0.0372	

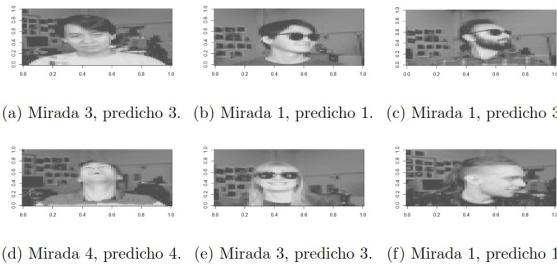


Figura 14: Clasificación de dirección de la mirada de los sujetos, correspondientes a una red neuronal de 1 capa oculta con 2 neuronas ocultas y función sigmoide logística. “izquierda”=1, “derecha”=2, “de frente”=3, “arriba”=4. Fuente: Elaboración propia.

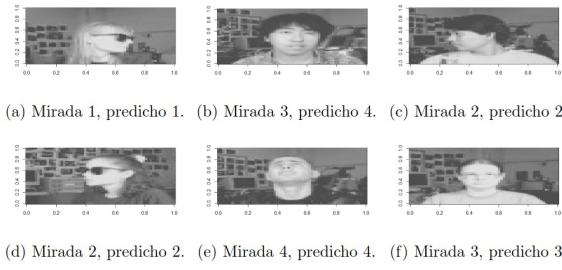


Figura 15: Clasificación de dirección de la mirada de los sujetos, correspondientes a una red neuronal de 1 capa oculta con 2 neuronas ocultas y función sigmoide arcotangente. “izquierda”=1, “derecha”=2, “de frente”=3, “arriba”=4. Fuente: Elaboración propia.

Al considerar una neurona oculta las proporciones de error con ambas funciones sigmoide son significativamente elevadas. No obstante, a partir de 2 neuronas ocultas las proporciones de error son iguales para ambas funciones sigmoide y además son muy pequeñas, por lo que es recomendable utilizar una red neuronal de una capa oculta con al menos 2 neuronas ocultas y cualquiera de las dos funciones sigmoide para clasificar la dirección de la mirada con estos datos.

Para terminar, en las Figuras 14 y 15 se muestran ejemplos de cómo la red neuronal clasifica la dirección de la mirada de los sujetos.

### 3.2.4. Clasificación de emociones

Finalmente, para clasificar las emociones de los sujetos, se etiquetaron las emociones de la siguiente manera: “enojado”=1, “feliz”=2, “neutral”=3 y “triste”=4. Se ajustaron redes neuronales usando la función sigmoide logística y la función sigmoide arcotangente. En cada caso se consideró una capa oculta y se varió el número de neuronas ocultas de 1 a 6. En las tablas 15–18 se presentan las tablas de confusión correspondientes a 1 y 6 neuronas ocultas.

Después de obtener las tablas de confusión se obtuvieron las proporciones de error que se presentan en la Tabla 19.

Las proporciones de error con ambas funciones sigmoide son significativamente elevadas con una neurona oculta. No obstante, a partir de dos neuronas ocultas, es recomendable utilizar la función sigmoide logística debido a que su proporción de error de clasificación, en general, es menor que la correspondiente a la función sigmoide arcotangente. Sin embargo, a partir de tres neuronas ocultas con la función sigmoide arcotangente se tienen proporciones de error de clasificación muy pequeñas.

Tabla 15: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 1 neurona oculta.

		Predichos			
		1	2	3	4
Reales	1	55	0	0	0
	2	24	20	2	0
3	1	5	13	21	
4	0	0	0	47	

Tabla 17: Tabla de confusión de la red neuronal con función sigmoide logística de 1 capa oculta con 6 neuronas ocultas.

		Predichos			
		1	2	3	4
Reales	1	55	0	0	0
	2	0	44	1	1
3	0	2	37	1	
4	0	0	0	47	

Tabla 16: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 1 neurona oculta.

		Predichos			
		1	2	3	4
Reales	1	55	0	0	0
	2	24	20	2	0
3	1	4	14	21	
4	0	0	0	47	

Tabla 18: Tabla de confusión de la red neuronal con función sigmoide arcotangente de 1 capa oculta con 6 neuronas ocultas.

		Predichos			
		1	2	3	4
Reales	1	55	0	0	0
	2	1	44	0	1
3	0	2	38	0	
4	0	0	0	47	

Tabla 19: Proporciones de error de clasificación considerando una capa oculta y variando de una a seis neuronas ocultas.

		Función sigmoide	
		Logística	Arcotangente
Neuronas ocultas	1	0.2819	0.2766
	2	0.0638	0.1543
3	0.0319	0.0745	
4	0.0266	0.0319	
5	0.0213	0.0372	
6	0.0266	0.0213	

Finalmente, en las figuras 16 y 17 se muestran ejemplos de cómo la red neuronal clasifica las emociones de los sujetos.



(a) Emoción 4, predicho 4. (b) Emoción 3, predicho 3. (c) Emoción 1, predicho 1.  
 (d) Emoción 3, predicho 4. (e) Emoción 2, predicho 2. (f) Emoción 1, predicho 1.

Figura 16: Clasificación de las emociones de los sujetos, correspondientes a una red neuronal de 1 capa oculta con 6 neuronas ocultas y función sigmoide logística. “enojado”=1, “feliz”=2, “neutral”=3, “triste”=4. Fuente: Elaboración propia.



(a) Emoción 1, predicho 1. (b) Emoción 4, predicho 4. (c) Emoción 3, predicho 3.  
 (d) Emoción 2, predicho 2. (e) Emoción 4, predicho 4. (f) Emoción 2, predicho 1.

Figura 17: Clasificación de las emociones de los sujetos, correspondientes a una red neuronal de 1 capa oculta con 6 neuronas ocultas y función sigmoide arcotangente. “enojado”=1, “feliz”=2, “neutral”=3, “triste”=4. Fuente: Elaboración propia.

## 4. CONCLUSIONES

En este trabajo, mediante redes neuronales artificiales, aplicando una función sigmoide logística o arcotangente y variando el número de neuronas ocultas dentro de una capa oculta, se clasificaron patrones encontrados en la literatura, y se evaluaron las redes neuronales mediante sus tablas de confusión y sus proporciones de error de clasificación.

En el ejemplo presentado de los datos de los dígitos escritos manualmente, tomados del conjunto de datos del Modified National Institute of Standards and Technology (MNIST), se concluye que puede utilizarse una red neuronal con una capa oculta, con al menos 10 neuronas ocultas y cualquiera de las dos funciones sigmoide para obtener una buena regla de clasificación con estos datos. Si se desea tener proporciones de error más pequeñas, se tendría que agregar más neuronas ocultas dentro de la capa oculta.

En el ejemplo de la base de datos de las imágenes de rostros de varias personas en varias poses, se concluye que para estos datos puede utilizarse una red neuronal de una capa oculta, con una sola neurona y cualquiera de las dos funciones sigmoide para obtener una buena regla de clasificación en la identificación de un sujeto de interés, así como también para la clasificación de sujetos con gafas de sol. Para la clasificación de la dirección de las miradas, se concluye que puede utilizarse una red neuronal de una capa oculta, con al menos 2 neuronas ocultas y cualquiera de las dos funciones sigmoide. En el caso de la clasificación de las emociones, se obtiene una buena regla de clasificación con una red neuronal de una capa oculta, con al menos 3 neuronas ocultas y cualquiera de las dos funciones sigmoide.

Se observó, en los ejemplos de aplicación presentados en este trabajo, un buen comportamiento de las redes neuronales artificiales para clasificar patrones. También vimos en estos ejemplos que es posible tener una buena clasificación considerando una sola capa oculta y pocas neuronas ocultas. Cuando el número de clases aumenta, por lo general se requieren más neuronas ocultas para tener una proporción de error de clasificación

baja.

## Referencias

- Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press, London, England.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.
- Devroye, L., Györfi, L. & Lugosi G. (2013). *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York.
- Dixit, R., Kushwah, R. & Pashine, S. (2020). Handwritten digit recognition using machine and deep learning algorithms. *International Journal of Computer Applications*, 176(42), 27–33.
- García Molina, J., Herrero, J., de Jesús, A., López, J., Guisado, M.A., Bustamante, Á. & Padilla, W. (2018). *Ciencia de datos: técnicas analíticas y aprendizaje estadístico*. Alfaomega, Bogotá.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York.
- Izenman, A. (2013). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer, New York.
- Mitchell, T. (1997). *Machine Learning*. McGrawHill Education, New York.
- Ripley, B. & Venable, W. (2021). Package “nnet”. Disponible en: <https://cran.r-project.org/web/packages/nnet/nnet.pdf>
- Rodríguez-Flores, M. A. (2010). Desarrollo de un sistema basado en un clasificador neuronal para reconocimientos de orugas. Tesis presentada en la Universidad Nacional Autónoma de México. Disponible en: [http://132.248.9.195/ptd2010/abril/0656324/0656324\\_A1.pdf](http://132.248.9.195/ptd2010/abril/0656324/0656324_A1.pdf)
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.