Study of the Effect of Combining Activation Functions in a Convolutional Neural Network

M. Guevara, V. Cruz, O. Vergara, *Senior Member, IEEE*, M. Nandayapa, *Member*, *IEEE*, H. Ochoa, *Member*, *IEEE*, and H. Sossa, *Senior Member, IEEE*

***Abstract*—Convolutional Neural Networks (CNN's) have proven to be an effective approach for solving image classification problems. The output, the accuracy and the computational efficiency of a CNN are determined mainly by the architecture, the convolutional filters, and the activation functions. Based on the importance of an activation function, in this paper, nine new activation functions based on combinations of classical functions such as ReLU and sigmoid are presented. Also, a study about the effects caused by the activation functions in the performance of a CNN is presented. First, every new function is described, also, their graphs, analytic forms and derivatives are presented. Then, a traditional CNN model with each new activation function is used to classify three 10-class databases: MNIST, Fashion MNIST and a handwritten digit database created by us. Experimental results illustrate that some of the proposed activation functions lead to better performances on classifying than classical activation functions. Moreover, our study demonstrated that the accuracy of a CNN could be increased by 1.18% with the new proposed activation functions.**

***Index Terms*—Activation Function, Convolutional Neural Network, Modified National Institute of Standards and Technology, Fashion Modified National Institute of Standards and Technology, Sigmoid, Rectified Linear Unit.**

I. Introducción

Las Redes Neuronales Convolucionales (CNN's, por sus siglas en inglés) son un conjunto de algoritmos modelados a partir del cerebro humano, diseñados para reconocer patrones, y para resolver problemas de clasificación de imágenes. La exactitud y la eficiencia computacional de una CNN son determinadas principalmente por la arquitectura y los filtros convolucionales; sin embargo, para su desempeño existe un elemento fundamental: la función de activación.

María Cristina Guevara Neri es estudiante del Programa de Doctorado en Ciencias en Ingeniería Avanzada de la Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México, al171517@alumnos.uacj.mx.

Vianey Cruz, Osslan Vergara, Manuel Nandayapa, y Humberto Ochoa, son profesores de la Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México, {vianey.cruz, overgara, mnandaya, hochoa}@uacj.mx.

Juan Humberto Sossa Azuela, es profesor del Instituto Politécnico Nacional (CIC-IPN), Ciudad de México, México, humbertosossa@gmail.com.

La función de activación es una parte crucial de un sistema de clasificación (o reconocimiento) que eventualmente se utilizará para tomar una decisión respecto a si un objeto, una persona, un animal, o un dígito fue clasificado correctamente en determinada clase, de acuerdo con una perspectiva humana [1]. Como su nombre lo indica, la función de activación tiene la intención de lograr aquello que los humanos realizan implícitamente cada vez que algo necesita ser clasificado; esto es, activar la combinación de características correspondientes del objeto a identificar y clasificar, de manera que se pueda concluir el tipo de objeto que está siendo catalogado. Aunque la tarea de identificar es particularmente sencilla de llevar a cabo para el ojo humano, tal acción representa un reto más complicado para una máquina.

En la literatura se han publicado diversos artículos donde se estudian funciones de activación tales como las unidades que emplea un rectificador. Por ejemplo, en [2] se describe la aplicación de una unidad lineal rectificada (ReLU, por sus siglas en inglés) como función de activación en un sistema implementado con una red neuronal profunda para la clasificación de los dígitos manuscritos de la base de datos MNIST (Modified National Institute of Standards and Technology database, por sus siglas en inglés). Además, se realizó la comparación del desempeño de la función ReLU contra el de *softmax*, donde la primera fue superada por la segunda debido a un problema de convergencia.

Por otro lado, en [3] se propone una versión renovada de la función ReLU implementada en una CNN para resolver un problema de clasificación con la base de datos CIFAR-10, que contiene 60,000 imágenes a color de 10 clases diferentes de objetos que incluye automóviles, aviones, gatos, perros, entre otras. Además, se presenta una comparación del comportamiento de distintas versiones de la ReLU, tales como la unidad lineal exponencial (ELU, por sus siglas en inglés), donde se muestra cómo al efectuar modificaciones no lineales sobre la función ReLU se puede mejorar su rendimiento e incrementar su velocidad de convergencia.

Así mismo, en [4] se propone el uso de funciones de activación limitadas no sigmoideas para el reconocimiento de imágenes de la base de datos MNIST utilizando redes neuronales profundas, donde el comportamiento de la red en términos de la estabilidad del entrenamiento da como resultado una reducción significativa en la probabilidad de un problema de inestabilidad numérica.

Las variaciones de la función ReLU, como la función SELU (*Scaled Exponential Linear Unit*) descrita en [5], y las funciones EReLU (*Elastic Rectified Linear Unit*) y EPReLU (*Elastic Parametric Rectified Linear Unit*) utilizadas en [7], muestran que al modificar algunas características de las funciones se puede mejorar su comportamiento y consecuentemente los resultados generados.

Es importante mencionar que no sólo se utilizan

combinaciones algebraicas típicas como funciones de activación en un sistema de clasificación. Distintas funciones no algebraicas, como la familia de trascendentes, son empleadas frecuentemente como activación en una Red Neuronal Artificial (ANN, por sus siglas en inglés), desde funciones exponenciales hasta funciones trigonométricas.

En [5-8] se presenta una comparación entre funciones lineales y no lineales que son modificaciones de la función exponencial tales como la sigmoid y la tangente hiperbólica. Además, se demuestra que las funciones no lineales pueden tener un desempeño similar al de las lineales y ofrecen la ventaja de obtener variaciones en los valores de salida generados por la parte negativa de los valores de entrada (a diferencia de ReLU). Sin embargo, los cálculos realizados con funciones no lineales son generalmente más complejos que los hechos con las funciones lineales. No obstante, la función sigmoid y sus variaciones son utilizadas comúnmente. Por ejemplo, las funciones *Swish* (producto de la función lineal y la sigmoid) y la tangente hiperbólica penalizada mostradas en [8] reportan un comportamiento estable en tareas de procesamiento del lenguaje natural generando resultados equiparables a los obtenidos por las funciones lineales. Así mismo, la sigmoid linealizada propuesta en [9] demuestra que un comportamiento no lineal de la función de activación conlleva a una mejor representación de características.

Los resultados reportados en los trabajos [3-9] son una motivación para desarrollar el punto de estudio del presente artículo: combinaciones de funciones de distinta naturaleza, desde funciones algebraicas comunes hasta variaciones de funciones trascendentes para crear nueve funciones de activación nuevas. Sin embargo, existen diferencias notables entre la propuesta que en el presente artículo se ofrece y los trabajos que forman parte de la literatura, las cuales se resumen de la siguiente manera:

1) La mayoría de los estudios hechos son llevados a cabo en redes neuronales profundas, mientras que en nuestro trabajo se utiliza una CNN tradicional.
2) Para el problema de reconocimiento de imágenes se emplean tres bases de datos distintas donde se utilizan dos conjuntos ampliamente conocidos: MNIST y Fashion MNIST. No obstante, cabe resaltar que por lo general ambos conjuntos no suelen incluirse juntos en el mismo estudio. Además, se incluye una tercera base de datos donde el grupo de imágenes de dígitos manuscritos tiene la característica de que el área del dígito no está normalizada en tamaño sobre el fondo, a diferencia de las imágenes de MNIST.
3) Las funciones de activación propuestas resultan de la unión de dos funciones que pertenecen a distintas familias. Es decir, a diferencia por ejemplo de la función *swish* que resulta de una operación aritmética entre dos funciones o de la tangente hiperbólica penalizada que es el escalamiento de la función trigonométrica original, nuestra combinación conserva las características propias de cada tipo de función por separado con el fin de aprovechar tanto el fácil manejo de operaciones aritméticas dentro de las funciones algebraicas, como el rango acotado de las trascendentes. Aclarando en este punto que también incluimos dos combinaciones hechas únicamente entre funciones trascendentes.

El enfoque principal de los experimentos se centra en la evaluación del desempeño de las nuevas funciones de activación implementadas cada una de ellas por separado en una CNN tradicional, y la comparación de los resultados de todas las funciones empleadas, tanto las funciones creadas como las comúnmente utilizadas. El objetivo es demostrar que el desempeño del clasificador se verá afectado por los cambios en la forma de la función de activación, y su rendimiento será enriquecido al combinar las características de dos funciones distintas.

El presente trabajo se encuentra organizado como se detalla a continuación. En la sección II se presenta la descripción general de una CNN y se explica la función de activación. En la sección III se definen los materiales y métodos utilizados. En la sección IV se presentan los experimentos y los resultados obtenidos. Por último, las conclusiones y los trabajos a futuro son mostrados en la sección V.

II. Red Neuronal Convolucional y Funciones de Activación

Desde que las CNNs fueron introducidas por Yann LeCunn y sus colaboradores en los años noventa [10], [11] han demostrado su excelente desempeño en diversas tareas, tales como reconocimiento de imágenes [12] o clasificación de señales [13] por mencionar algunas. Una CNN es una red neuronal multicapa entrenada con un algoritmo de propagación hacia atrás (*back-propagation*), diseñada para reconocer patrones visuales directamente desde imágenes con mínimo preprocesamiento.

Actualmente, existen diversas arquitecturas de CNNs, tales como AlexNet [12], LeNet-5 [14], ZF Net [15], GoogLeNet [16], Microsoft ResNet [17], entre otras. Sin embargo, la estructura básica de una CNN se compone de cinco capas [4]: 1) entrada, 2) convolución, 3) reducción (*pooling*), 4) completamente conectada, y 5) clasificación (*softmax*).

En la capa de entrada se especifica el tamaño de la imagen: altura, ancho y número de planos (para una imagen en escala de grises es uno y para una imagen en color es tres en representación de los valores RGB).

La capa de convolución es la más importante debido a que no sólo se aprenden las características de la imagen de entrada, sino que se pueden analizar a profundidad los componentes de la CNN para posteriormente formar un mapa de características. La idea es transformar la imagen en una representación que pueda ser procesada de manera más sencilla sin que pierda sus principales características para diferenciarla de otra.

La operación de convolución se realiza con una matriz denominada núcleo, filtro o neurona. Para determinar los valores de los filtros, denominados coeficientes o pesos, éstos deben ser aprendidos durante el proceso de entrenamiento de toda la red, donde posteriormente los coeficientes resultantes se convolucionan con la imagen de entrada. Generalmente los filtros de la capa de entrada extraen características de bajo

nivel –como texturas y bordes–, y conforme la señal avanza hacia la salida, las capas extraen otro tipo de características. El filtro se aplica sobre un conjunto de pixeles de la imagen y su tamaño debe ser menor al tamaño de la imagen, e igual al tamaño del conjunto de pixeles (llamado *patch*) sobre los cuales está siendo aplicado.

La operación de convolución consiste en realizar el producto punto entre el filtro y el *patch*. El filtro se mueve paso a paso (según sea configurado el *stride*) de manera horizontal por toda la imagen, empezando desde la esquina superior izquierda hasta llegar a la esquina inferior derecha. Cada paso dado por el filtro genera un valor como resultado del producto punto que es almacenado en una nueva matriz la cual se conoce como mapa de activación.

Los coeficientes de los filtros asignan importancia a diversos aspectos en la imagen, de manera que eventualmente puedan ayudar a diferenciarla de otra. Los pesos suelen ser variables ajustables y son reforzados por otros valores ajustables llamados bias, los cuales son independientes de las entradas. La ecuación **(1)** muestra la operación de convolución.

$$\sum_i w_i \, x_i + b. \tag{1}$$

donde w representa los pesos, b los bias, y x las entradas a los filtros. Después de la convolución, la red contiene un elemento crucial: la función de activación.

La función de activación contribuye a decidir si la salida de la ecuación **(1)** es suficiente para activar o no una neurona (considerando el hecho de que la salida de la neurona puede ser cualquier valor real). Existen algunas funciones de activación comúnmente utilizadas; las de la familia de las funciones algebraicas, específicamente de la familia de unidades lineales rectificadoras (Fig. 1a-c), y las de la familia de funciones transcendentes, particularmente del conjunto de funciones exponenciales (Fig. 1d-e).

Una función que es regularmente empleada en CNN's para problemas de clasificación, y que pertenece a la familia de unidades lineales rectificadoras es la función ReLU (Fig. 1a), cuya característica principal es generar una salida nula para cualquier entrada negativa (o entrada igual a cero) y conservar para cada entrada positiva su valor en la salida. La función ReLU es preferida en aplicaciones de redes neuronales profundas ya que agiliza el tiempo de entrenamiento por ser una función simple.

Existen dos variaciones de ReLU donde la salida de las entradas negativas es también negativa. La primera llamada *leaky* ReLU (LReLU) que se muestra en la Fig. 1b, afecta a las entradas negativas con un factor de 0.01, mientras que la segunda denominada *parametric* ReLU o PReLU mostrada en la Fig. 1c, afecta a las entradas negativas con un valor positivo variable y aprendible. Si bien las tres funciones son distintas entre sí, su derivada es un valor constante: positivo para las entradas positivas, y no positivo para las entradas no positivas.

Las funciones sigmoid (Fig. 1d) y tangente hiperbólica (tanh) (Fig. 1e) son exponenciales, con un rango de 0 a 1 para la primera, y de -1 a 1 para la segunda. La función tanh tiene la característica de ser impar mientras que su derivada es par. La derivada de la función sigmoid es una función par, lo cual puede ocasionar inconvenientes desde una perspectiva matemática ya que toda entrada positiva y su inverso aditivo generarán el mismo valor de salida.

La capa de convolución (y la función de activación) está seguida por una operación para reducir el tamaño del mapa de características y remover información espacial redundante, conocida como la capa de *pooling*, donde la operación más popular es la denominada *maxpooling* que consiste en utilizar un filtro con un *stride* de igual longitud y aplicarlo a la entrada, donde la salida será igual al valor máximo de cada subregión donde el filtro es aplicado.

Enseguida de la capa de *pooling* se encuentra la capa completamente conectada (o más de una), donde las neuronas se conectan a todas las neuronas de la capa precedente.

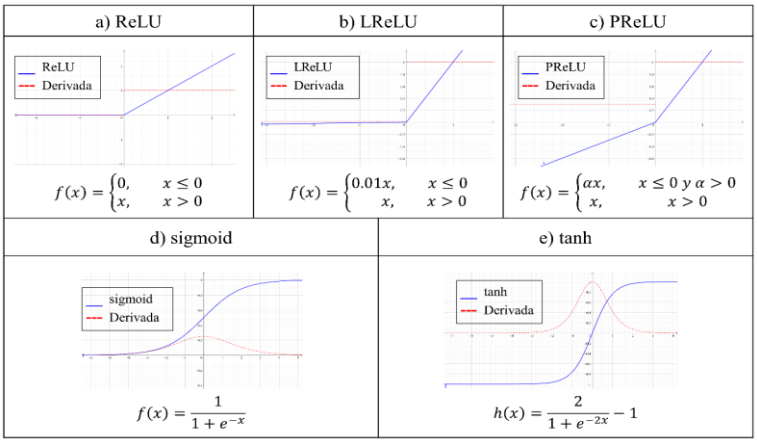


Fig. 1. Funciones de activación comúnmente utilizadas en problemas de clasificación de imágenes.

Las características de las imágenes aprendidas en las capas anteriores son combinadas para identificar los patrones más importantes. La última capa completamente conectada combina las características para clasificar las imágenes (en este punto las imágenes iniciales ya no son visualmente reconocibles). El número de salidas de la capa es igual al

número deseado de clases de datos.

En la última capa, la función softmax normaliza la salida de la capa completamente conectada y su salida consiste en valores positivos cuya suma es igual a 1, por lo cual las salidas son las probabilidades de pertenencia de la imagen de entrada a cada clase.

III. Materiales y Métodos

En la presente sección, se describen las bases de datos utilizadas. Posteriormente, se discute la estructura de la CNN implementada y la configuración de los parámetros de cada una de sus capas. Por último, se muestra la composición de las funciones propuestas, y la representación gráfica y analítica del resultado de las combinaciones realizadas.

A. Bases de Datos

Para los experimentos se seleccionaron tres bases de datos diferentes. Los detalles de los tres conjuntos se muestran a continuación.

MNIST. Fue creada por Y. LeCun y ha sido utilizada en la literatura como punto de referencia para resolver problemas de reconocimiento óptico de caracteres y probar la efectividad de algoritmos de aprendizaje de máquina. Está compuesta por imágenes de 70,000 dígitos manuscritos en escala de grises, con una resolución de $28x28$ [18]. El área del dígito es de tamaño $20x20$ y está centrada en el área de $28x28$ de acuerdo con el centro de masa de los pixeles. El subconjunto de entrenamiento contiene 60,000 ejemplos de los cuales 10,000 son para el subconjunto de prueba. Para los experimentos realizados se emplearon 1,000 dígitos seleccionados aleatoriamente, del 0 al 9 con 100 dígitos por clase. El subconjunto de entrenamiento contiene 750 ejemplos, y 250 son para el subconjunto de prueba. La Fig. 2a muestra un ejemplo correspondiente a la clase cero.

Fashion MNIST. Es un conjunto que se ha vuelto importante para comparar algoritmos de reconocimiento de patrones al igual que MNIST ya que ambas son estructuralmente similares. Está compuesta por imágenes de 28×28 pixeles de 70,000 prendas de vestir de 10 tipos de clases - 0: camiseta, 1: pantalón, 2: suéter, 3: vestido, 4: saco, 5: zapato, 6: blusa, 7: tenis, 8: bolsa, 9: bota - en escala de grises, con una resolución de $20x20$. El subconjunto de entrenamiento contiene 60,000 ejemplos, y 10,000 son para el subconjunto de prueba. Para los experimentos realizados se emplearon 1,500 imágenes seleccionadas aleatoriamente, 1,000 imágenes para entrenamiento, y 500 imágenes como casos de validación. La Fig. 2b muestra un ejemplo de un elemento de la base de datos, correspondiente a la clase 7: tenis.

Dígitos manuscritos (Propia). Se encuentra compuesta por imágenes de 1,000 dígitos, del 0 al 9 con 100 dígitos por clase, en formato binario con una resolución de $28x28$. La gran diferencia contra MNIST es que los dígitos no se encuentran centrados y el área que ocupan puede ser de diversos tamaños. El subconjunto de entrenamiento contiene 750 ejemplos, y 250 son para el subconjunto de prueba. Para los experimentos realizados se empleó la base de datos completa. La Fig. 2c muestra un ejemplo de un elemento de la base de datos, correspondiente a la clase dos.

a) b) c)

Fig. 2. Ejemplos de elementos de las bases de datos.

B. Estructura de la CNN

Para los experimentos se decidió implementar una CNN, cuyo modelo se muestra en la Fig. 3, tomando para su descripción como ejemplo de entrada una imagen del conjunto MNIST. Cada capa se describe a continuación.

ENTRADA. El tamaño de la imagen es de 28×28 pixeles.

Conv_1. Se aplican 8 filtros de 5×5 pixeles a la entrada, con un *stride* de 1 y un relleno (*padding*) de 0. La salida es de tamaño $24 \times 24 \times 8$, donde 24×24 es el tamaño de la imagen convolucionada y 8 el número de mapas de características producido por cada filtro.

Act_Func. Las funciones de activación utilizadas serán descritas, cada una de ellas a detalle, en la subsección III-C.

MaxPool_1. Se define un tamaño de 2×2 para la región rectangular de reducción, con un *stride* de 2. La salida es de tamaño $12 \times 12 \times 8$, donde 12×12 es el tamaño de los datos reducidos y 8 es el número de mapas de características.

Conv_2. Se aplican 16 filtros de 5×5, con un *stride* de 1 y un *padding* de 0. La salida es de tamaño $8 \times 8 \times 16$.

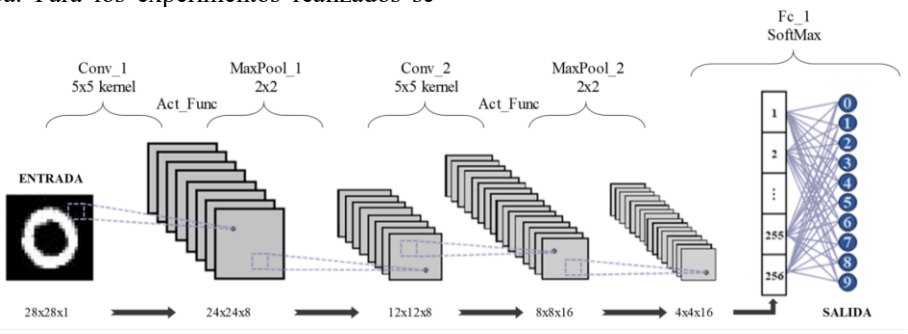


Fig. 3. Estructura implementada de la CNN.

Maxpool_2. Se define un tamaño de 2×2 para la región rectangular de reducción, con un *stride* de 2. La salida es de tamaño $4 \times 4 \times 16$.

Fc_1. Tiene 256 entradas las cuales son el resultado del producto matemático de la salida dada por la capa previa ($4 \times 4 \times 16 = 256$). El tamaño de la salida es 10, que es igual

al número deseado de clases (una por clase).

SoftMax. La salida es un vector de tamaño 10×1, el cual está formado por valores positivos (cuya suma es igual a 1).

C. *Funciones de Activación*

Para el estudio se propusieron nueve funciones de activación nuevas: una función obtenida de la modificación de la sigmoid, llamada sigmoid2, y ocho funciones obtenidas de la combinación de la parte negativa de una función y la parte positiva de otra. A continuación, se presenta una descripción de la función sigmoid2.

SIGMOID2. Es una función trascendente, y una versión modificada de la sigmoid. Los cambios en sigmoid2 se observan en amplitud y desplazamiento, y en realidad la asemejan más a la tanh, la cual tiene la característica de proporcionar salidas negativas para cualquier entrada negativa, y salidas positivas para cualquier entrada positiva.

Para formar las combinaciones se utilizó la función sigmoid2, así como tres diferentes funciones que son tradicionalmente utilizadas como activación: ReLU, PReLU y tanh. En las combinaciones de los diversos tipos de funciones se condicionaron dos cosas: la función resultante no debe ser par, y su dominio no debe tener restricción para ningún valor de entrada a la red.

Es primordial reiterar que las funciones generadas son distintas a las mostradas en la Fig. 1, es decir, son funciones cuya composición combina características de las funciones conocidas. A continuación, se describe brevemente cada una de las ocho funciones nuevas obtenidas por combinaciones.

SIGMOID2_TANH. Es una combinación de la parte negativa de la función sigmoid2, y la parte positiva de la función tanh.

TANH_SIGMOID2. Es una combinación de la parte negativa de la función tanh, y la parte positiva de la función sigmoid2.

0_SIGMOID2. Es una combinación de la parte negativa de la ReLU, y la parte positiva de sigmoid2.

0_TANH. Es una combinación de la parte negativa de la función ReLU, y la parte positiva de la función tanh.

ALPHA_SIGMOID2. Es una combinación de la parte negativa de la función PReLU, y la parte positiva de la función sigmoid2.

ALPHA_TANH. Es una combinación de la parte negativa de la función PReLU, y la parte positiva de la función tanh.

SIGMOID2_X. Es una combinación de la parte negativa de la función sigmoid2, y la parte positiva de la función ReLU.

TANH_X. Es una combinación de la parte negativa de la función tanh, y la parte positiva de la función ReLU.

La Fig. 4 muestra las expresiones algebraicas de las funciones de activación propuestas, incluyendo su forma visual y su derivada.

Salida. Después de que el conjunto de probabilidades es obtenido, la clase con la mayor probabilidad de pertenencia es activada y la imagen es clasificada.

IV. Experimentación y Resultados

Todos los experimentos fueron realizados en una laptop HP, con un procesador Intel Core i5 y memoria RAM de 8Gb. Además, se utilizó el software MATLAB para implementar los algoritmos de aprendizaje, y el *Deep Learning Toolbox*.

Las funciones de activación fueron implementadas una por una con los conjuntos de entrenamiento y validación de las tres bases de datos.

.

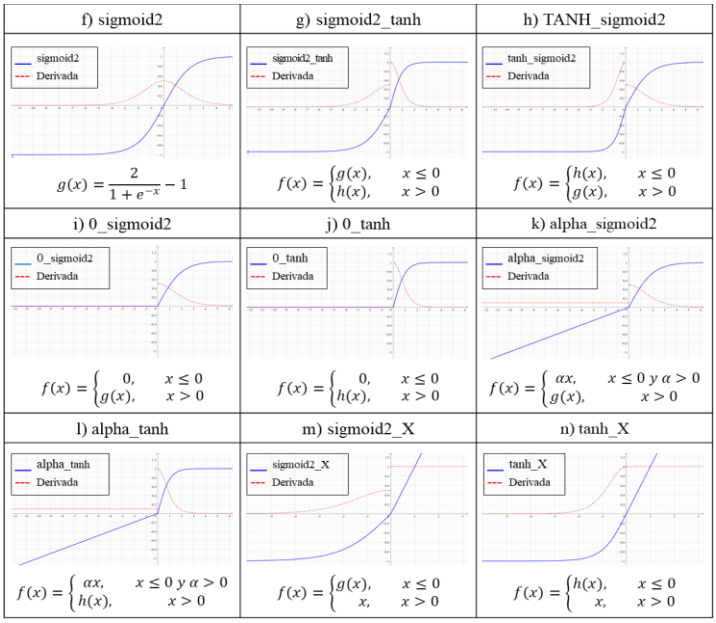


Fig. 4. Funciones de activación propuestas utilizadas en la CNN.

Las condiciones de aplicación de la CNN para cada experimento no fueron modificadas, es decir, lo único que cambió para cada prueba fue la función de activación. Adicionalmente al cambio de los valores de los pesos, los cuales son inicializados aleatoriamente en cada ejecución, se mantuvo un *learning rate* de 0.01 y para la minimización se

utilizó el algoritmo de optimización del Gradiente Descendiente Estocástico con Momento (SGDM, por sus siglas en inglés).

En total se realizaron 20 corridas de la CNN para cada función y se calculó la exactitud en cada una de ellas con el fin de promediar los valores y evaluar qué tan frecuente fue correcta la clasificación. Después, se comparó numéricamente el valor promedio de la exactitud de cada una de las funciones de activación para determinar cuáles ofrecieron el mejor desempeño para cada conjunto de datos

En la Fig. 5 se muestran los resultados de exactitud para las catorce funciones por cada base de datos utilizada, donde están resaltados los cinco mejores resultados generados por funciones clásicas (PReLU, ReLU y LReLU), así como por funciones propuestas (sigmoid2_X, tanh_X, 0_sigmoid2 y alpha_sigmoid2). Es importante observar que las funciones proporcionaron valores de exactitud muy cercanos entre ellos. Además, en la Tabla I, se muestra un resumen de los resultados obtenidos por el clasificador con las cinco mejores funciones de activación, donde se puede apreciar que de manera general entre las siete funciones con los mejores desempeños se encuentran cuatro de las combinaciones propuestas, cada una de ellas formada por funciones de familias distintas.

Para el conjunto MNIST, el mejor desempeño obtenido (función LReLU) fue de 228 imágenes clasificadas correctamente de un conjunto de 250, y el quinto mejor desempeño logrado (función ReLU) fue de 226 imágenes clasificadas de manera correcta. En la base de datos Fashion MNIST, el mejor desempeño obtenido (función tanh_X) fue de 379 imágenes clasificadas correctamente de un conjunto de 500, y el quinto mejor desempeño logrado (función ReLU) fue de 373 imágenes clasificadas de manera correcta. Cabe destacar que tres de los mejores desempeños corresponden a funciones de activación propuestas. Para el conjunto propio de dígitos manuscritos, el mejor desempeño obtenido (función LReLU) fue de 236 imágenes clasificadas correctamente de un conjunto de 250, y el quinto mejor desempeño (función PReLU) logrado fue de 232 imágenes clasificadas de manera correcta.

Los resultados de exactitud fueron estudiados de manera más profunda, por lo cual se obtuvo el desempeño de las funciones de la Fig. 5, por clase y para cada base de datos, para conocer cuál logró obtener el mejor desempeño, no sólo de manera general sino por clase. Los resultados se muestran en las Tablas II-IV, donde los valores más altos por clase fueron resaltados con negritas y los más bajos fueron subrayados. Es importante resaltar que los resultados no son tan distintos entre sí.

De las Tablas II-IV se puede notar que en las tres bases de datos algunas funciones conocidas como ReLU, PReLU y LReLU generaron un buen desempeño, al igual que las funciones propuestas como la sigmoid2 y sus combinaciones. Así mismo, se debe recordar que las funciones propuestas combinan las características propias de cada una de las partes, por lo cual, funciones como la 0_sigmoid2 lleva consigo el problema de desvanecimiento del gradiente (es decir, el valor de la derivada se vuelve nulo con el tiempo).

Aun cuando los resultados buscados son aquellos donde los valores numéricos son los más altos; los resultados más bajos indican que los cambios hechos en la función de activación sí generan un efecto en la exactitud obtenida por la CNN. De acuerdo con las características de cada función utilizada, y con base a los mejores resultados obtenidos (Tablas II-IV), se debe resaltar que para intentar obtener mejores resultados de exactitud, las combinaciones propuestas deben evitar el problema de desvanecimiento del gradiente.

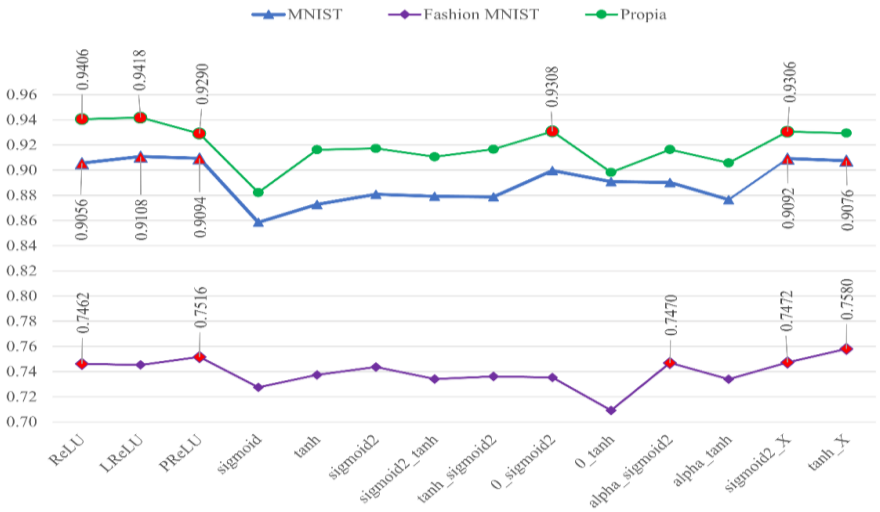


Fig. 5. Exactitudes obtenidas con las 14 funciones de activación en la clasificación de las tres bases de datos de prueba.

Las combinaciones que no presentan dicho problema generaron los mejores desempeños en la clasificación de cada conjunto de datos, con excepción de la función 0_sigmoid2 que sí presenta desvanecimiento del gradiente y aun así generó resultados competitivos en la clasificación de la base de datos propia.

Como se observa en la Fig. 5 y en las Tablas I-IV las funciones de activación propuestas para nuestro análisis presentan un desempeño equiparable al mostrado por las funciones tradicionales aun cuando no son mejores en todas las comparaciones. Sin embargo, como la propia literatura lo señala, la tarea de encontrar una función de activación cuyo rendimiento sea el mejor de manera universal se encuentra aún lejos de ser una realidad. Por tanto, es importante resaltar como un éxito el hecho de que las exactitudes obtenidas utilizando funciones propuestas sean mayores en algunas de las clasificaciones realizadas.

TABLA I
COMPARACIÓN DE LOS CINCO MEJORES RESULTADOS DE EXACTITUD PARA CADA BASE DE DATOS

Función de Activación	MNIST	Fashion MNIST	PROPIA
LReLU	**0.9108**	-	**0.9418**
PReLU	0.9094	0.7516	0.9290
ReLU	0.9056	0.7462	0.9406
tanh_X	0.9076	**0.7580**	-
sigmoid2_X	0.9092	0.7472	0.9306
alpha_sigmoid2	-	0.7470	-
0_sigmoid2	-	-	0.9308

TABLA II
EXACTITUD POR CLASE DEL CONJUNTO DE VALIDACIÓN PARA LAS MEJORES FUNCIONES DE ACTIVACIÓN CON LA BASE DE DATOS MNIST

	FUNCIÓN DE ACTIVACIÓN				
CLASE	ReLU	LReLU	PReLU	Sigmoid2_X	tanh_X
0	0.9780	0.9640	**0.9820**	0.9800	0.9800
1	0.9560	0.9640	0.9560	**0.9720**	0.9600
2	0.8740	0.8740	0.8700	0.8620	**0.8840**
3	0.9100	0.8980	0.9020	0.9000	**0.9120**
4	**0.9040**	0.8880	0.8940	0.8920	0.8780
5	0.9100	**0.9420**	0.9100	0.9120	0.9020
6	**0.9440**	0.9360	0.9420	0.9420	0.9380
7	0.8680	0.8820	0.8840	**0.8940**	0.8760
8	0.8720	0.8740	**0.9040**	0.8720	0.8680
9	0.8400	**0.8860**	0.8500	0.8660	0.8780

TABLA III
EXACTITUD POR CLASE DEL CONJUNTO DE VALIDACIÓN PARA LAS MEJORES FUNCIONES DE ACTIVACIÓN CON LA BASE DE DATOS FASHION MNIST

	FUNCIÓN DE ACTIVACIÓN				
CLASE	ReLU	PReLU	alpha_ sigmoid2	Sigmoid 2_X	tanh_X
0	0.7730	0.7980	**0.8010**	0.7660	0.7820
1	0.9280	0.9280	0.9290	**0.9390**	0.9270
2	0.5180	0.4820	0.5620	**0.5860**	0.5560
3	0.7650	0.7770	0.7920	0.7590	**0.8040**
4	0.6800	**0.7140**	0.6920	0.6750	0.7020
5	0.8750	0.8790	0.8740	0.9000	**0.9110**
6	0.3020	**0.3250**	0.2690	0.2850	0.2960
7	**0.8430**	0.8380	0.8320	0.8380	0.8340
8	**0.8580**	0.8280	0.8070	0.8010	0.8310
9	0.9200	**0.9470**	0.9120	0.9230	0.9250

TABLA IV
EXACTITUD POR CLASE DEL CONJUNTO DE VALIDACIÓN PARA LAS MEJORES FUNCIONES DE ACTIVACIÓN CON LA BASE DE DATOS DE DÍGITOS PROPIA

	FUNCIÓN DE ACTIVACIÓN				
CLASE	ReLU	LReLU	PReLU	0_sigmoid2	Sigmoid2_X
0	**0.9700**	0.9660	**0.9700**	0.9660	0.9680
1	**0.9360**	0.9320	0.9060	0.9200	0.9180
2	**0.9600**	0.9460	0.9500	0.9380	0.9420
3	**0.9480**	0.9260	0.9340	0.9380	0.9360
4	**0.9300**	0.9220	0.9200	0.8860	0.8760
5	**0.9100**	0.9080	0.9000	0.9080	0.8960
6	0.9440	**0.9520**	0.9240	0.9280	0.9240
7	0.9060	0.9460	0.8900	0.9060	**0.9500**
8	0.9520	0.9540	0.9500	**0.9680**	0.9340
9	0.9500	**0.9660**	0.9460	0.9500	0.9620

Adicionalmente, en la Tabla I, se puede observar que para MNIST no se supera la máxima exactitud reportada en la literatura (0.9977), y lo mismo sucede para Fashion MNIST (0.9920), sin embargo, los resultados son competitivos. Se debe recordar que en el presente trabajo se implementó una CNN clásica para los tres conjuntos de datos. A diferencia de las arquitecturas con las que se han reportado los mejores resultados, que son muy profundas y que fueron artesanalmente diseñadas para cada conjunto de datos en particular. Por lo tanto, para el caso de la generalización de la arquitectura de una CNN, se observa que las funciones de activación, efectivamente, tienen un efecto en su desempeño y que las combinaciones propuestas pueden mejorar hasta en un 1.18% la exactitud como se observa en Fashion MNIST. Para el caso de MNIST la mejor combinación propuesta se queda por debajo en un 0.32% contra el mejor caso, y para la base de datos propia la diferencia es de tan solo 1.12%. Finalmente, se observa que para cada clase de cada conjunto de datos las diferencias se encuentran en promedio entre 0.6 y 1.18%.

Por lo anterior, los aportes del presente trabajo fueron: 1) se combinaron diferentes tipos de funciones y se aplicaron como función de activación en una CNN para clasificar 3 conjuntos diferentes de imágenes, 2) se mostró la variación del desempeño del clasificador al modificar la función de activación con las diferentes combinaciones propuestas, 3) se obtuvieron resultados similares entre sí con las funciones propuestas y las comúnmente utilizadas, incluso algunos valores de exactitud con funciones propuestas fueron mayores que los obtenidos con las funciones conocidas, 4) se demostró que se puede expandir el conjunto de funciones que pueden ser utilizadas para resolver un problema de clasificación de imágenes con resultados prometedores.

V. CONCLUSIONES

En el presente trabajo se mostró una comparación del desempeño de una CNN utilizando catorce funciones de activación, en un problema de clasificación de dígitos manuscritos y de artículos de moda. Para los experimentos, se utilizaron cinco funciones de activación ampliamente conocidas, y se propusieron nueve combinaciones nuevas de las funciones, en donde cada una se aplicó por separado.

De acuerdo con los resultados del clasificador, se puede concluir que sí es posible realizar combinaciones de funciones

de activación usualmente utilizadas con el fin de crear una función distinta y posteriormente emplearla como función de activación en una CNN, ya que se lograron obtener resultados semejantes (e incluso mejores en algunos casos) que los generados por funciones conocidas, tales como la función ReLU (y sus variaciones) y la función sigmoid. Por lo anterior, se puede concluir que las funciones de activación propuestas tienen un efecto positivo cuando son añadidas al conjunto de funciones clásicas utilizadas en un problema de clasificación.

Como trabajo futuro, se pretenden expandir los experimentos a problemas de clasificación que vayan más allá del reconocimiento de dígitos e imágenes en general en escala de grises, con el fin de comparar el desempeño de las funciones de activación comúnmente utilizadas con el de las funciones de activación propuestas. Así mismo, se encuentra como trabajo posterior fortalecer la arquitectura de la red, añadir más capas de convolución e incrementar el tamaño de los filtros utilizados, con la intención de mejorar los resultados de clasificación obtenidos en el presente trabajo.

AGRADECIMIENTOS

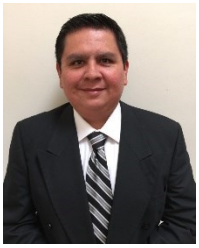
Los autores agradecen a la Universidad Autónoma de Ciudad Juárez y al Instituto Politécnico Nacional por el apoyo brindado. H. Sossa agradece al IPN y al CONACYT por el apoyo económico brindado realizar la presente investigación en el marco de los apoyos 20190007 y 65, respectivamente.

REFERENCIAS

[1] S. Qian, H. Liu, C. Liu, S. Wu, and H. San, "Adaptive activation functions in convolutional neural networks", Neuromputing, vol. 272, pp. 204-212, 2018.
[2] A. Agarap, "Deep learning using rectified linear units (relu)", Preprint, pp. 1-7, https://arxiv.org/abs/1803.08375, 2018.
[3] L. Guifang and S. Wei, "Research on convolutional neural network based on improved Relu piecewise activation function", Procedia Computer Science, vol. 131, pp. 977-984, 2018.
[4] S. Liew, M. Khalil-Hani and R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems", Neurocomputing, vol. 216, pp. 718-734, 2016.
[5] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task", Preprint, pp. 1-5, https://arxiv.org/abs/1804.02763, 2018.
[6] X. Jiang, Y. Pang, X. Li, and J. Pan, "Deep neural networks with elastic rectified linear units for object recognition", Neurocomputing, vol. 275, pp. 1132-1139, 2018.
[7] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning", arXiv preprint arXiv:1811.03378, 2018.
[8] S. Eger, P. Youssef and I. Gurevych, "Is it time to swish? Comparing deep learning activation functions across NLP tasks", arXiv preprint arXiv:1901.02671, 2019.
[9] V. Singh and V. Kumar, "Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability", Expert Systems with Applications, vol. 120, pp. 346-356, 2019.
[10] Y. LeCun, B. Boser, J. Denker, and D. Henderson, "Backpropagation applied to handwritten zip code recognition", Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.
[11] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network", Advances in Neural Information Processing Systems, vol. 2, pp. 396-404, 1990.
[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", Communications of the ACM, pp. 84-90, vol. 60, no. 6, 2017.
[13] A. Suárez-León and J. Núñez, "1D Convolutional neural network for detecting ventricular heartbeats", IEEE Latin America Transactions, vol. 17, no. 12, pp. 1970-1977, 2019.
[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.
[15] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks", Proceedings of European Conference on Computer Vision (ECVV), vol. 8689, pp. 818-833, 2014.
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and D. Erhan, "Going deeper with convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.
[17] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition", Preprint, pp. 770-778, https://arxiv.org/abs/1512.03385, 2016.
[18] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database", AT&T Labs, 2010. Available: http://yann. lecun. com/exdb/mnist 2.

María Cristina Guevara Neri was born in Ciudad Juárez, Chihuahua, México, on March 17, 1988. She received the B.S. degree in electromechanical engineering from the Instituto Tecnológico de Ciudad Juárez, México, in 2012; and the M. S. degree in electric engineering at the Universidad Autónoma de Ciudad Juárez. Currently, she is studying the PH.D. on advanced engineering.

Vianey Guadalupe Cruz Sánchez was born in Cárdenas, Tabasco, México, on September 14, 1978. She earned the B.S. degree in computer engineering from the Instituto Tecnológico de Cerro Azul, México, in 2000; the M.Sc. degree in computer science at the Center of Research and Technological Development (CENIDET) in 2004; and the Ph.D. in computer science from CENIDET in 2010. She currently serves as a professor at the Autonomous University of Ciudad Juarez, Chihuahua, México. She is a member of the IEEE Computer Society. Her fields of interest include neuro symbolic hybrid systems, digital image processing, knowledge representation, artificial neural networks and augmented reality.



Osslan Vergara (SM'12), was born in Cuernavaca, Morelos, Mexico on July 3, 1977. He earned the BS degree in Computer Engineering from the Instituto Tecnologico de Zacatepec, Mexico, in 2000; the MSc in Computer Science at the Center of Research and Technological Development (CENIDET) in 2003; and the PhD degree in Computer Science from CENIDET in 2006. He currently serves as a professor at the Universidad Autónoma de Ciudad Juárez, Chihuahua, Mexico, where he is the head of the Computer Vision and Augmented

Reality laboratory. Prof. Vergara is a level one member of the Mexican National Research System. He serves several peer-reviewed international journals and conferences as editorial board member and as a reviewer. He has coauthored more than 100 book chapters, journals, and international conference papers. Dr. Vergara has directed more than 50 BS, MSc, and PhD thesis. He is a senior member of the IEEE Computer Society and member of the Mexican Computing Academy. His fields of interest include pattern recognition, digital image processing, augmented reality and mechatronics.

Manuel Nandayapa received a B.S. degree in Electronics Engineering from Institute of Technology of Tuxtla Gutierrez, Chiapas, Mexico in 1997, M.S. degree in Mechatronics Engineering from CENIDET, Morelos, Mexico in 2003, and D.Eng degree in energy and environmental science from the Nagaoka University of Technology, Japan, in 2012. His research interests include mechatronics, motion control, and haptic interfaces. He is with the Department of Industrial and Manufacturing Engineering at Autonomous University of Ciudad Juarez. Dr. Nandayapa is Member of the IEEE Industrial Electronics Society and Robotics Automation Society.

Humberto de Jesus Ochoa received the B.Eng. degree in industrial electronics from the Technological Institute of Veracruz, México, his M.Sc. in electronics from the Technological Institute of Chihuahua, México, and Ph.D. degree in electrical engineering from the University of Texas at Arlington, USA. He is currently with the Department of Ingeniería Eléctrica y Computación at the Universidad Autónoma de Ciudad Juárez, Mexico. He worked as an Electronic Officer for the Mexican Merchant Marine. His current teaching and research interests include multirate systems for medical image analysis, images restoration and reconstruction, image and video coding, statistical signal processing and pattern recognition.

Juan Humberto Sossa Azuela received his BS degree in Communications and Electronics from the University of Guadalajara in 1980. He obtained his Master's degree in electrical engineering from CINVESTAV-IPN in 1987 and his PhD in Informatics form the INPG, France in 1992. He is currently a full-time professor at the Robotics and Mechatronics Laboratory of the Center for Computing Research of the National Polytechnic Institute from Mexico since 1996. He has more than 450 journal and conference publications. He is a Senior Member of the IEEE.