

Metaheurísticas y CNN: Comparación de modelos híbridos para mejorar la clasificación de imágenes

Gerardo Treviño-Valdés¹, Jesús Alejandro Navarro-Acosta²,
Marco Antonio Aceves-Fernández¹, Jesús Carlos Pedraza-Ortega¹,
Saúl Tovar-Arriaga¹

¹ Universidad Autónoma de Querétaro,
Facultad de Ingeniería,
México

² Universidad Autónoma de Coahuila,
Centro de Investigación en Matemáticas Aplicadas,
México

{geratrevino115, alexnav24, marco.aceves}@gmail.com
caryoko@yahoo.com, saul.tovar@uaq.mx

Resumen. En este artículo presenta la comparación entre una red neuronal convolucional estándar y una variante híbrida de la misma incluyendo algoritmos metaheurísticos para la clasificación de imágenes. Se pretende mejorar la clasificación del algoritmo implementando un ajuste en los hiperparámetros de la red neuronal convolucional con el algoritmo híbrido propuesto. Los algoritmos metaheurísticos aquí presentados son: Algoritmos Genéticos, Optimización por enjambre de partículas, Optimización de Lobo Gris y Optimización de Ballena Jorobada. Los resultados muestran que la metodología implementada es capaz de aumentar la exactitud en la clasificación, como de algunas otras de las métricas para evaluar la clasificación.

Palabras clave: Red neuronal convolucional, hibridismo, metaheurísticas, clasificación de imágenes.

Metaheuristics and CNN: Hybrid Model Comparison to Improve Image Classification

Abstract. In this paper presents the comparison between a standard convolutional neural network and a hybrid variant of it, including metaheuristic algorithms for image classification. The aim is to improve the classification of the algorithm by implementing an adjustment in the hyperparameters of the convolutional neural network with the proposed hybrid algorithm. The metaheuristic algorithms presented here are: Genetic Algorithms, Particle Swarm Optimization, Gray Wolf Optimization, and Humpback Whale Optimization. The results show that the methodology implemented is able for increasing the accuracy in the classification, as well as some other of the metrics to evaluate the classification.

Keywords: Convolutional neural network, hybridism, metaheuristics, image classification.

1. Introducción

Las redes neuronales convolucionales (CNN) han demostrado un rendimiento en el procesamiento de imágenes al aumentar su desempeño en tareas como clasificación, detección de objetos, entre otras. Así como la capacidad de adaptación de sus modelos [5]. Los enfoques de detección de objetos basados en características y clasificadores de aprendizaje automático han sido muy fructíferos hasta tiempos recientes.

Cuando se aplican a diferentes tareas o se adaptan para desafíos adicionales, estos requieren de un ajuste de parámetros y una reducción dimensional para lograr un rendimiento aceptable. Las redes neuronales convolucionales, en los últimos años, han propiciado un importante avance en tareas que involucran visión artificial, tales como clasificación, localización, detección y segmentación de objetos, descripción de escenas, entre otras, ya sea en imágenes o vídeo.

Los resultados que se obtienen actualmente se puedan emplear en una gran variedad de aplicaciones. Sin embargo, el desempeño de algoritmos como las CNN en la detección de objetos depende en gran medida de la elección y ajuste de diversos hiperparámetros que pueden determinar la tasa de aprendizaje de las mismas, por tal motivo en este artículo se presenta la combinación de dos técnicas de inteligencia artificial como lo son los algoritmos metaheurísticos y las redes neuronales convolucionales para realizar un entrenamiento más robusto en comparación del entrenamiento estándar de estas redes [13].

Y de esta forma lograr un modelo eficiente y eficaz para la clasificación de objetos. Con el fin de ajustar los hiperparámetros de la CNN se implementan y comparan cuatro metaheurísticas ampliamente utilizados en el estado del arte como son optimización por enjambre de partículas (PSO), algoritmos genéticos (GA), optimización de lobos grises (GWO) y optimización de ballena jorobada (WOA).

2. Marco teórico

2.1. Redes neuronales convolucionales

Las redes convolucionales, son un tipo especializado de red neuronal para procesar datos que tiene una topología similar a una cuadrícula (matriz). Las redes convolucionales han tenido un éxito extraordinario en aplicaciones prácticas. Las redes convolucionales son simplemente redes neuronales que usan la convolución en lugar de la multiplicación general de matrices en al menos una de sus capas.

Estas son muy potentes para todo lo que tiene que ver con el análisis de imágenes, debido a que son capaces de detectar características simples como por ejemplo detección de bordes, líneas, etc. Y extraer características más complejas hasta llegar a su objetivo.

Consta de capas convolucionales y de reducciones alternadas, y en sus capas finales tiene capas de conexión total como una red perceptrón multicapa Figura 1. En la convolución se realizan operaciones de productos y sumas entre la capa de partida y los n filtros que genera un mapa de características. Las características extraídas corresponden a cada posible ubicación del filtro en la imagen original [6].

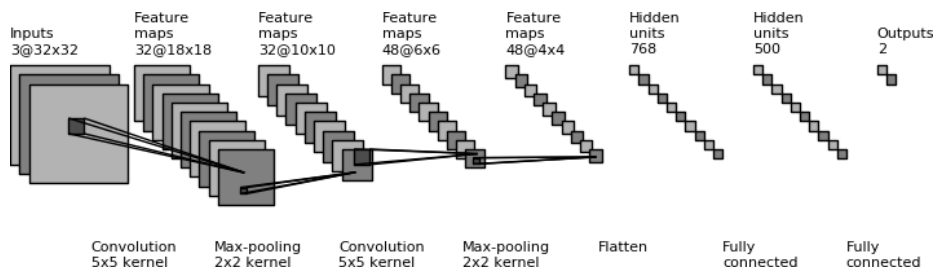


Fig. 1. Arquitectura de las CNN [6].

La ventaja es que el mismo filtro sirve para extraer la misma característica en cualquier parte de la entrada, con esto que consigue reducir el número de conexiones y el número de parámetros a entrenar en comparación con una red multicapa de conexión total [2].

2.2. Metaheurísticas

Las técnicas de optimización metaheurísticas se han inspirado principalmente en conceptos muy simples. Estos algoritmos suelen basarse en fenómenos físicos, comportamientos de animales o conceptos evolutivos. Tienen mayor flexibilidad a diferentes problemas sin ningún cambio especial en la estructura del algoritmo, ya que en su mayoría asumen los problemas como cajas negras.

En otras palabras, solo las entradas y salidas de un sistema son importantes para una metaheurística [7]. Las metaheurísticas tienen capacidades superiores para evitar los óptimos locales en comparación con las técnicas de optimización convencionales.

Esto se debe a la naturaleza estocástica de las metaheurísticas que les permiten evitar el estancamiento en las soluciones locales y adentrarse extensamente en todo el espacio de búsqueda. El cual para problemas reales suele ser desconocido o muy complejo y con una gran cantidad de óptimos locales, por lo que las metaheurísticas tienen buen desempeño únicamente teniendo claro el objetivo [11].

Algoritmos genéticos (GA). El algoritmo genético es una metaheurística inspirada en el proceso de selección natural creado por John Henry Holland en el año 1970, surgió con este algoritmo base de muchas representaciones metaheurísticas [15]. Un algoritmo genético estándar requiere dos requisitos previos, es decir, una representación genética del dominio de la solución y una función de aptitud para evaluar a cada individuo.

La idea central del algoritmo genético es permitir que los individuos evolucionen a través de algunas operaciones genéticas como se muestra en el algoritmo. Las operaciones populares incluyen selección, mutación, cruce. El proceso de selección nos permite preservar a los individuos fuertes mientras eliminamos a los débiles. Las formas de realizar la mutación y el cruce a menudo se basan en las propiedades del problema específico [8].

Optimización por enjambre de partículas (PSO). El algoritmo de optimización con enjambre de partículas (PSO) fue desarrollado por J. Kennedy y R. C. Eberhart [3], el cual se basa del comportamiento de parvadas de aves, colonias de abejas, bancos de peces, entre otros.

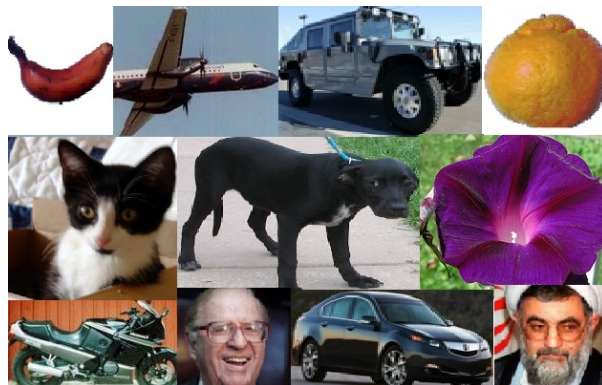


Fig. 2. Muestra de la base de datos Natural Images[12].

Se puede utilizar para resolver problemas de optimización que carecen de conocimiento del dominio. La población está constituida por una serie de partículas. Cada uno de ellas representa un individuo. Busca la mejor solución actualizando velocidad y vector de partículas de acuerdo con las ecuaciones (1) y (2). Donde v_{id} representa la velocidad de la partícula i en la d -ésima dimensión, x_{id} representa la posición de la partícula i . P_{id} y P_{gd} son los mejores locales y el mejor global, r_1 , r_2 son números aleatorios entre 0 y 1, mientras que w , c_1 y c_2 son peso de inercia y coeficientes de aceleración para explotación y aceleración para los coeficiente de exploración, respectivamente:

$$V_{id}(t + 1) = w * v_{id}(t) + c_1 * r_1 * (P_{id} - x_{id}(t)) + c_2 * r_2 * (P_{gd} - x_{id}(t)), \quad (1)$$

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1). \quad (2)$$

Optimizador lobo gris (GWO). El algoritmo metaheurístico del lobo gris salió a la luz en 2014 por obra de Seyedali Mirjalili [10]. Donde se muestra el comportamiento de este animal su forma de caza y su conducta social y de particular interés es que tienen una jerarquía social dominante muy estricta donde llamamos a estos grupos como alfa, beta, delta y omega, cada una de estos grupos juega un papel importante en la manada.

Para modelar matemáticamente la jerarquía social de los lobos, consideramos la solución más adecuada como la alfa (a). En consecuencia, la segunda y tercera mejores soluciones se nombran beta (b) y delta (d) respectivamente. Se supone que el resto de las soluciones candidatas como omega (x).

En el algoritmo GWO la búsqueda está guiada por a , b y d . Los lobos x siguen a estos tres lobos y así estos rodean a sus presas durante la caza. Matemáticamente se representa en la ecuación (3) y (4), donde t indica la iteración actual, \vec{A} y \vec{C} son vectores de coeficientes, \vec{X}_p es el vector de posición de la presa, \vec{X} indica el vector de posición de un lobo gris:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \quad (3)$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \quad (4)$$

Tabla 1. Clases y numero de imágenes del dataset Natural Images [12].

Clase	Nombre	Imágenes por clase
1	Avión	727
2	Automóvil	968
3	Gato	885
4	Perro	702
5	Flor	843
6	Fruta	1000
7	Motocicleta	788
8	Persona	986

donde los componentes de \vec{a} se reducen linealmente de 2 a 0 en el transcurso de las iteraciones y r_1, r_2 son vectores aleatorios en $[0, 1]$:

$$\vec{A} = 2\vec{a} \cdot r_1 - \vec{a}, \tag{5}$$

$$\vec{C} = 2 \cdot r_2. \tag{6}$$

Algoritmo de optimización de ballenas (WOA). El algoritmo de optimización de la ballena jorobada se presenta en 2017 por Seyedali Mirjalili [9]. Se puede interpretar como una modificación al algoritmo del lobo gris (GWO) donde en este caso representa de igual manera su comportamiento de caza, las ballenas jorobadas pueden reconocer la ubicación de sus presas y rodearlas.

Las ecuaciones principales son las descritas en el algoritmo GWO a diferencia del este método, una ecuación en espiral es creado entre la posición de la ballena y la presa para imitar el movimiento en forma de hélice de las ballenas jorobadas dada la siguiente ecuación (9):

$$\vec{X}(t + 1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \tag{7}$$

donde \vec{D}' indica la distancia de la ballena a la presa (la mejor solución obtenida hasta ahora), b es una constante para definir la forma de la espiral logarítmica, l es un valor aleatorio de $[-1, 1]$ y \cdot es una multiplicación elemento por elemento.

Aquí se tiene en cuenta el vector donde una ballena crea un círculo que se contrae para llegar a su presa, se asume una probabilidad del 50 por ciento para elegir esta distinción al modelo circular GWO:

$$\vec{X}(t + 1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{si } p < 0,5, \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{si } p \geq 0,5, \end{cases} \tag{8}$$

donde p es un número aleatorio uniforme de $[0,1]$.

Algoritmo 1. Algoritmo híbrido CNN-Metaheurístico

```
1: Preprocesamiento ← clases Imágenes
2: modelo ← Se crea el modelo CNN;
3: while i ≤ epoch do
4:   Época CNN ← Ejecución de una época en el entrenamiento CNN
5:   get weights → hiperparámetros
6:   Inicio Metaheurístico:
7:   Parámetros iniciales. ← it
8:   Ingreso de hiperparámetros al algoritmo ← se evita empeorar la salida
9:   while j ≤ it do
10:    Generación aleatoria de individuos
11:    Ejecución
12:    Aptitud de los individuos ←  $\text{Max } f(x) = w1 * Ex + w2 * F1$ 
13:    Mejores individuos
14:  end while
15:  return Hiperparámetros
16:  set weights ← hiperparámetros
17: end while
18: modelo entrenado
19: Evaluación
20: return clasificación de imagen
```

3. Materiales y métodos

3.1. Metaheurístico adaptado para el ajuste fino de hiperparámetros CNN

El objetivo del algoritmo en este artículo es aprovechar el beneficio de la búsqueda exhaustiva de los algoritmos metaheurísticos sobre los hiperparámetros de la red neuronal convolucional centrándose en las capas densas donde se trabaja con una extensa cantidad de hiperparámetros.

Las capas convolucionales en los algoritmos de clasificación como detección de objetos son de suma relevancia. Estas capas juegan un papel importante en la parametrización de las entradas (imágenes) y dan paso a las capas densas para realizar las operaciones necesarias y llegar a la clasificación. Los algoritmos metaheurísticos tienen mecanismos libres de derivación.

A diferencia de los enfoques de optimización basados en gradientes que por su practicidad se utilizan en redes neuronales y redes neuronales convolucionales. Las metaheurísticas optimizan estocásticamente los problemas. El proceso de optimización comienza con soluciones aleatorias y no es necesario calcular la derivada de los espacios de búsqueda para encontrar el óptimo. Esto hace que la metaheurística sea adecuada para problemas reales con información desconocida o compleja [1].

Se tiene claro que la combinación de estos métodos puede realizarse de infinitas formas, por ejemplo: utilizar el metaheurístico después de cada etapa de entrenamiento o época, en determinadas épocas del entrenamiento, únicamente para su ajuste final, por mencionar algunas. Todo esto se determina arbitrariamente dependiendo de la complejidad de la arquitectura, efectividad de convergencia, el tiempo de entrenamiento, entre otras [14].

Tabla 2. Clases y numero de imágenes del dataset.

Capa	Kernel	Parámetros
Conv2d	3×3	640
Max Pooling	2×2	-
Conv2d	3×3	36928
Max Pooling	2×2	-
Flatten	-	-
Dense1	32	663584
Dense2	32	1056
Dense3	8	264

En este caso se examina únicamente el algoritmo descrito en el pseudocódigo del Algoritmo 1 donde al finalizar cada época se ejecutan los algoritmos de optimización para el ajuste fino de estos hiperparámetros sumado las operaciones que se ejecutan a través de las épocas en la CNN. Se genera un punto de partida para homogeneizar las condiciones iniciales de cada algoritmo híbrido iniciando con una época base para todos los algoritmos comparados.

3.2. Base de datos

La base de datos Natural Images empleada en este artículo consta de 6899 imágenes distintas divididas en 8 clases diferentes [12]. En la Figura 2 se muestra una imagen representativa de las imágenes a trabajar observamos que son de diferentes tamaños y estilos. Las clases que contiene el dataset se muestran en la Tabla 2, con su respectivo nombre y el número de imágenes por clase, para poder emplear esta base de datos se toma el total de imágenes y aplica un pequeño pre-procesamiento al normalizar a 80×80 píxeles.

3.3. Implementación del algoritmo híbrido

Parámetros. La arquitectura utilizada en este artículo se basa en la propuesta original ilustrada en la Figura 1 y se representa en la Tabla 2 de manera detallada, similar a la LeNet-5 propuesta por Yann LeCun [6]. Se realizará el ajuste de hiperparámetros en las capas densas durante 10 épocas de entrenamiento. Por lo tanto, los algoritmos descritos requieren parámetros de inicialización para su funcionamiento.

Cada uno de los metaheurísticos se estableció con 30 individuos, 10 iteraciones por cada época de la red neuronal convolucional, una ventana de 25, y una longitud del problema igual a la cantidad de hiperparametros a evaluar después de las capas convolucionales en este caso particular sería de 664,896 variables.

3.4. Función objetivo

En este trabajo se propone una función objetivo a maximizar (Ec. 9) por parte de los enfoques híbridos presentados. Esta, integra la exactitud Ex (Ec. 10) y la medición F1-score $F1$ (Ec. 13), las cuales son dos de las métricas más utilizadas en el campo de la clasificación:

Tabla 3. Evaluación de los algoritmos.

Método	Exactitud	Precisión	Sensitividad	F1-score
CNN	0.8202	0.9962	0.1178	0.2054
CNN-GA	0.8224	0.9318	0.0901	0.1605
CNN-PSO	0.8260	0.9740	0.1079	0.1911
CNN-GWO	0.8355	0.9740	0.2450	0.3864
CNN-WOA	0.8347	0.9956	0.4403	0.6053

$$\text{Max } f = w1 * Ex + w2 * F1, \quad (9)$$

donde $w1 = 0.4$ y $w2 = 0.6$, son ponderaciones de la importancia de dichas métricas. Estos valores se obtuvieron mediante pruebas exhaustivas donde se observó que la CNN presenta mejor desempeño usando dicha configuración.

Para fines de comparación, en los resultados se presentan por separado los valores de Exactitud (Ex), F1-score, así como los de precisión (Pre) (Ec. 11) y sensitividad (Sens) (Ec. 12). Métricas que componen la F1-Score [4]. Donde TP es Verdadero Positivo (abreviado por sus siglas en inglés), TF es Verdadero Negativo, FP es Falso Positivo y FN corresponde a Falso Negativo:

$$Ex = \frac{TP + TF}{TP + TF + FP + FN}, \quad (10)$$

$$Pre = \frac{TP}{TP + FP}, \quad (11)$$

$$Sens = \frac{TP}{TP + FN}, \quad (12)$$

$$F1 = \frac{2 \times Pre \times Sens}{Pre + Sens} = \frac{2 \times TP}{2 \times TP + FP + FN}. \quad (13)$$

4. Resultados experimentales y discusión

Una vez implementada la metodología propuesta se obtienen los datos presentados en la Tabla 3 para los máximos valores alcanzados por los 4 algoritmos híbridos y la CNN estándar de las métricas antes descritas. El entrenamiento de la red neuronal convolucional se realizó con una relación 80/20 en los datos de entrenamiento y prueba, se realizaron 10 distintos experimentos hasta alcanzar los máximos valores que podemos apreciar en dicha tabla.

Comparando los modelos híbridos con el método CNN obtenemos que en la precisión los mejores resultados fueron logrados por el método estándar de la CNN con un diferencia de 0.06 % al segundo mejor y del 6.44 % al peor. La sensitividad fue mejorada por CNN-WOA en 32.25 % respecto a CNN. En la evaluación F1-score se destaca nuevamente WOA con una diferencia de 40.0 %. Finalmente para la exactitud se mejoró por el algoritmo CNN-GWO un 1.53 % al modelo estándar CNN.

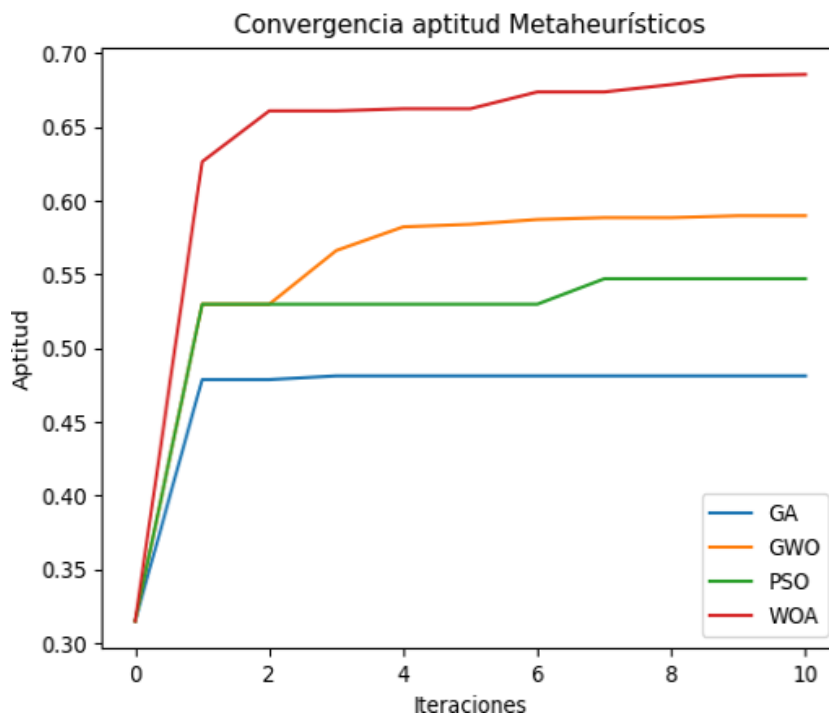


Fig. 3. Comparación de convergencia de métodos metaheurísticos, 1 de 10 épocas.

Como se mostró en los resultados al modificar la arquitectura de la CNN utilizando la metodología híbrida propuesta en este trabajo se observa una mejora en algunas de las métricas pero la más notoria es F1-score lo cual nos indica que el utilizar algoritmos híbridos de inteligencia artificial puede mejorar la robustez del modelo final por lo tanto mejorar la clasificación de imágenes.

Analizando los resultados obtenidos podemos concluir que algunos de los resultados en los modelos híbridos no superaron el modelo estándar, esto podría deberse a la naturaleza de los mismos, debido a las pocas iteraciones en las pruebas realizadas. Y se concluye para la tarea de clasificación aquí presentada el algoritmo con mejor desempeño es CNN-WOA.

En la Figura 1 se representa gráficamente la convergencia al mayor valor de exactitud obtenido en la experimentación a través de las épocas establecidas para cada uno de los métodos. El comportamiento de los algoritmos durante las épocas se intercala utilizando el método CNN y el algoritmo metaheurístico. Podemos notar que existen fluctuaciones en estos cambios siendo la más notoria en CNN-PSO, a través de las épocas cae la puntuación y en la siguiente iteración es recuperada.

Esto podría deberse a la elección de los hiperparámetros de la entrada a la CNN, no logra aprovechar las propiedades del algoritmo híbrido por sus bajas iteraciones en la etapa metaheurística. En la Figura 3 observamos un ejemplo de la convergencia en la primera época de la CNN evaluada por los metaheurísticos, desde esta instancia se aprecia como destaca el algoritmo WOA.

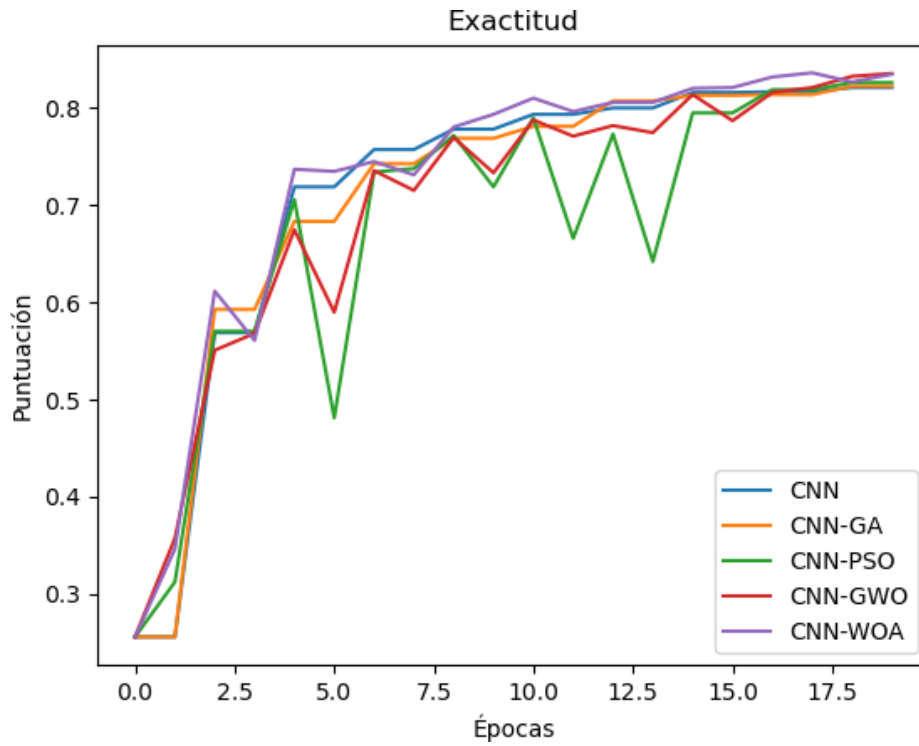


Fig. 4. Gráfica sobre épocas de la Exactitud.

5. Conclusiones y trabajo futuro

A lo largo de esta investigación se analizó el uso de metaheurísticas para el ajuste de hiperparámetros en las redes neuronales convolucionales. Dichos modelos híbridos presentan una mejora considerable para algunas de las métricas utilizadas en el campo de la clasificación como sensibilidad, precisión, exactitud y F1. Siendo esta última una de las más relevantes en la clasificación de imágenes por medio de CNNs la cual demostró una diferencia significativa favorable en el modelo CNN-WOA.

La metodología aquí implementada se aventaja de la búsqueda exhaustiva de estos algoritmos de optimización ampliando el espacio de búsqueda. En este trabajo de investigación se experimentó con la intercalación de capas utilizando el método CNN y el algoritmo de optimización con el fin de mejorar el desempeño de la red por época. Como trabajo futuro se plantea realizar el ajuste únicamente en la última capa de la red neuronal convolucional implicando cambios relevantes de los hiperparámetros para mejorar la eficiencia general en la red neuronal.

Acceder a equipo de cómputo robusto en la cual se puedan realizar mayor número de iteraciones en los algoritmos híbridos presentados obteniendo mejor desempeño en las variantes GA y PSO. De igual forma se plantea la implementación de estos modelos híbridos en distintas arquitecturas complejas de las CNN, y aplicarlo en diversas áreas como lo son la detección de objetos en imágenes y vídeo.

Referencias

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: From natural to artificial systems. vol. 1, no. 1 (1999) doi: 10.1093/oso/9780195131581.001.0001
2. Cai, Z., Fan, Q., Feris, R. S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision, pp. 354–370 (2016) doi: 10.48550/ARXIV.1607.07155
3. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995) doi: 10.1109/mhs.1995.494215
4. Hossain, M., Sulaiman, M. N.: A review on evaluation metrics for data classification evaluations. International Journal of Data Mining and Knowledge Management Process, vol. 5, no. 2, pp. 1–11 (2015) doi: 10.5121/ijdkp.2015.5201
5. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
6. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324 (1998) doi: 10.1109/5.726791
7. Lones, M., Luke, S.: Essentials of metaheuristics (2011) doi: 10.1007/s10710-011-9139-0
8. McCall, J.: Genetic algorithms for modelling and optimisation. Journal of computational and Applied Mathematics, vol. 184, no. 1, pp. 205–222 (2005), doi: 10.1016/j.cam.2004.07.034
9. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in Engineering Software, vol. 95, pp. 51–67 (2016) doi: 10.1016/j.advengsoft.2016.01.008
10. Mirjalili, S., Mirjalili, S. M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software, vol. 69, pp. 46–61 (2014) doi: 10.1016/j.advengsoft.2013.12.007
11. Osman, I. H., Laporte, G.: Metaheuristics: A bibliography (1996) doi: 10.1007/bf02125421
12. Roy, P., Ghosh, S., Bhattacharya, S., Pal, U.: Effects of degradations on deep neural network architectures (2018) doi: 10.48550/ARXIV.1807.10108
13. Syulistyo, A. R., Purnomo, D. M. J., Rachmadi, M. F., Wibowo, A.: Particle swarm optimization (PSO) for training optimization on convolutional neural network (CNN). Jurnal Ilmu Komputer dan Informasi, vol. 9, no. 1, pp. 52–58 (2016) doi: 10.21609/jiki.v9i1.366
14. Wang, B., Xue, B., Zhang, M.: Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2020) doi: 10.48550/ARXIV.1907.12659
15. Weile, D. S., Michielssen, E.: Genetic algorithm optimization applied to electromagnetics: A review. IEEE Transactions on Antennas and Propagation, vol. 45, no. 3, pp. 343–353 (1997) doi: 10.1109/8.558650