

Análisis exploratorio del algoritmo DeepGA en MNIST y Fashion MNIST.

García Sanchez Eddi Alberto¹, Juárez Caballero Emmanuel Isaac¹

¹ Instituto de Investigaciones en Inteligencia Artificial, Universidad Veracruzana.
Xalapa, Ver.
{zS21000455, zS21000449}@estudiantes.uv.mx

Resumen: Se presenta una comparativa entre el algoritmo de neuro evolución DeepGA y redes neuronales convolucionales del estado del arte sobre los conjuntos de imágenes MNIST y Fashion-MNIST, obteniendo mejores resultados tanto de precisión como una menor cantidad del número de parámetros con el uso de DeepGA, con el cual se logra una precisión promedio del 98.63 ± 0.20 % para el conjunto MNIST y de 90.80 ± 0.53 % para el conjunto Fashion-MNIST. Se observa una ventaja en el porcentaje de clasificación en ambos conjuntos de datos mediante el uso de DeepGA correspondientes a valores que oscilan entre (0.29 % - 53.41 %) para MNIST y de (5.76 % - 42.11 %) para Fashion-MNIST. Mientras que el número de parámetros obtenido para el conjunto MNIST fue de 24104 ± 5188.54 y para Fashion-MNIST de 56687.6 ± 11929.7 .

Palabras clave: Neuro evolución, Redes Neuronales Convolucionales, MNIST, Fashion-MNIST.

1 Introducción

La introducción de las redes neuronales convolucionales en el campo de la visión por computadora ha tenido un desarrollo muy importante en los últimos años, tanto en el apartado del reconocimiento de imágenes [1] como en la clasificación de estas [2]. Particularmente, en la clasificación de imágenes, la invención de novedosas arquitecturas de redes neuronales convolucionales [1,2,3] han producido redes cuya efectividad en cuanto a precisión de clasificación se supera conforme surge una nueva arquitectura. No obstante, la gran mayoría de las arquitecturas más populares se componen de un gran número de parámetros que deben ser entrenados, lo que se traduce en un alto costo computacional al momento de entrenar una arquitectura. Es justamente este último punto el cual ha sido objeto de investigación en años recientes bajo el objetivo de desarrollar arquitecturas de redes neuronales convolucionales que posean una alta precisión de clasificación y a su vez la menor cantidad de parámetros posibles para reducir su costo computacional [4].

La neuro evolución permite la optimización de redes neuronales artificiales, desde la optimización de los pesos de la red hasta la evolución de las arquitecturas neuronales, mediante técnicas del cómputo evolutivo. Parte esencial de la neuro evolución en arquitecturas neuronales es el tipo de codificación que se emplea para representar a las redes neuronales dentro de los procesos de un algoritmo evolutivo, además del impacto que tiene la codificación sobre los grandes espacios de búsqueda. Actualmente, diversos tipos de codificación han sido propuestos y estudiados [5].

El presente trabajo representa una extensión al elaborado por Vargas et al. en [6], en el cual se propone una codificación del tipo híbrida para favorecer el descubrimiento de redes neuronales convolucionales competitivas en términos de

precisión y complejidad, esto acompañado de un algoritmo genético que en conjunto conforman el marco de neuro evolución, recibiendo por nombre DeepGA. El objetivo de este trabajo es evaluar el desempeño de DeepGA sobre bases de datos populares como lo son MNIST [7] y Fashion-MNIST [8], con el fin de analizar el comportamiento del algoritmo frente a arquitecturas ya bien conocidas como ResNet-18, VGG-16 y AlexNet.

2 DeepGA

Vargas et al. [6] proponen el algoritmo Deep Genetic Algorithm (DeepGA), el cual es capaz de manipular redes neuronales convolucionales que se representan mediante una codificación híbrida en donde se combinan la codificación del tipo bloque-encadenado (block-chained) y binaria. El Algoritmo Genético brinda el balance entre la explotación y exploración del espacio de búsqueda por medio de los operadores de cruce y mutación, mismos que son diseñados especialmente para operar sobre la ya mencionada codificación híbrida.

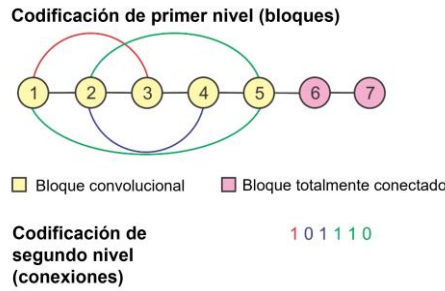


Fig. 1. Codificación empleada en DeepGA basada en bloques convolucionales simples [6].

El algoritmo DeepGA pretende maximizar el rendimiento de la clasificación y reducir la complejidad computacional de las redes neuronales convolucionales, entendiendo que la complejidad de estas es cuantificada con respecto al número de parámetros entrenables (valores de peso y sesgo). Para lograr este objetivo el DeepGA emplea una función de aptitud ponderada (Ec. 1) que contempla la precisión de clasificación del conjunto de validación y una relación de los parámetros de la red neuronal convolucional.

$$f(cnn) = (1 - w) * accuracy(cnn) + w * \frac{MP - NP}{MP} \quad (1)$$

donde cnn es la arquitectura de una red neuronal convolucional, w es el peso de penalización (establecido como 0.3), MP es el número máximo de parámetros permitidos en la red definido por el usuario y NP es el número de parámetros actuales de la arquitectura.

3 Metodología

Con el propósito de evaluar el funcionamiento de DeepGA en diversos conjuntos de imágenes se llevan a cabo una serie de adecuaciones a la implementación original propuesta por Vargas. et al. [6] y disponible en Github [9].

Entre las adecuaciones implementadas se realiza la detección de individuos dañados que no pertenecen al espacio de búsqueda. Los individuos identificados se desechan y en su lugar se selecciona aleatoriamente un individuo de la población. La operación del algoritmo DeepGA contemplando dicha modificación se describe en el siguiente pseudocódigo y se encuentra disponible en <https://github.com/EisaacJC/DeepGA-Analisis-Exploratorio->.

Input: Una población P de N individuos. El número de generaciones T , porcentaje de cruce $CXPB$, porcentaje de mutación $MUPB$ y tamaño del torneo $TSIZE$.
Output: Inicializar población (entrenamiento de las redes).
 $t \leftarrow 1$
while $t \leq T$ **do**
 Seleccionar $N/2$ padres por torneo probabilístico
 Offs $\leftarrow \{\}$
 while $|\text{Offs}| < N/2$ **do**
 Seleccionar dos padres aleatorios p_1 y p_2
 if $\text{random}(0, 1) \leq CXPB$ **then**
 $O_1, O_2 \leftarrow \text{Crossover}(p_1, p_2)$
 if $\text{random}(0, 1) \leq MUPB$ **then**
 Mutation(O_1, O_2)
 end if
 end if
 if O_1 es válido **then**
 fitness(O_1)
 else
 $O_1 \leftarrow$ Seleccionar un individuo aleatorio de la población
 end if
 if O_2 es válido **then**
 fitness(O_2)
 else
 $O_2 \leftarrow$ Seleccionar un individuo aleatorio de la población
 end if
 end while
 $P \leftarrow P \cup \text{Offs}$
 Seleccionar los mejores N individuos en P como sobrevivientes.
end while

Algoritmo 1. DeepGA.

Los conjuntos de imágenes elegidos para llevar a cabo los comparativos son MNIST [7] y Fashion-MNIST [8], ambos son conjuntos estándar para la realización de pruebas de rendimiento. En ambos casos la estructura de los conjuntos de datos es la misma: constan de 70,000 imágenes de 28x28 píxeles, las cuales se encuentran en escala de grises correspondientes a 10 clases balanceadas. En el caso de MNIST las clases son identificadas por números enteros que recaen en el rango de (0-9) [7] y las clases de Fashion-MNIST corresponden a tipos de ropa [8].

3.1 Configuración del experimento.

Para realizar la evaluación del algoritmo DeepGA en los conjuntos de datos mencionados previamente, se llevan a cabo 5 ejecuciones de 20 generaciones con los parámetros mostrados en la Tabla 1. La ejecución se realiza en el entorno GPU de Google Colaboratory en su versión de paga, la cual provee de una GPU Nvidia-P100 de 16 GB.

Para la evaluación de DeepGA se elige un porcentaje de 30% del conjunto de datos para hacer la validación y un 70% del mismo para llevar a cabo el entrenamiento.

Tabla 1. Elección de parámetros para las ejecuciones.

Parámetros	Valor
Porcentaje de cruza	70%
Porcentaje de mutación	70%
# Máximo de parámetros	2,000,000
Tamaño del torneo	5
Tasa de aprendizaje	0.0001
Tamaño del batch	256
Épocas	10

4 Resultados

El análisis de los resultados de DeepGA se realiza mediante la visualización del comportamiento de la función de aptitud, precisión y número de parámetros para las 5 ejecuciones en cada conjunto de datos (véase Fig 1 y Fig 2).

Se efectúa el estudio de los resultados de las 5 ejecuciones, esto es, se toma al individuo con mejor valor de aptitud de cada una de ellas para los dos conjuntos de datos y se estudian sus estadísticas correspondientes (véase Tabla 2 y Tabla 3).

Cabe mencionar que el tiempo de ejecución para cada una de las 5 ejecuciones realizadas fue de aproximadamente 8 horas con el hardware mencionado en la sección previa.

Conjunto de datos MNIST

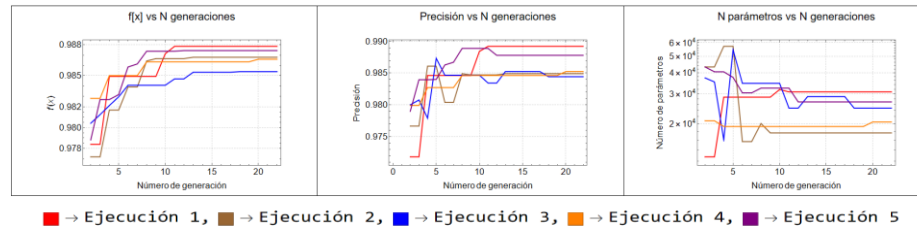


Fig. 2. Gráfico de la función de aptitud, precisión y número de parámetros para cinco ejecuciones del DeepGA en el conjunto de datos MNIST.

Tabla 2. Estadísticos correspondientes a cinco ejecuciones del algoritmo de Deep-GA para el conjunto de datos MNIST.

	Aptitud	Precisión	# de parámetros.
Mejor ejecución.	0.9878	0.9892	30794
Peor ejecución	0.9853	0.9844	24720
Promedio	0.9868	0.9863	24104
Desv. Estándar.	0.0009	0.0020	5188.54

Es posible observar a partir de la Tabla 2 y Fig 2 que el valor de la función de aptitud para la mejor y peor ejecución difieren únicamente en 0.0025, de igual forma debido al valor *cercano* a cero de la desviación estándar se observa que las ejecuciones son robustas, es decir, el comportamiento muestra una tendencia en los resultados a un

valor de aptitud cercano al máximo global, teniendo únicamente alrededor de 1.37% de error promedio en la precisión de clasificación y un valor promedio de 24104 parámetros para la clasificación en el conjunto MNIST.

Conjunto de datos: Fashion-MNIST.

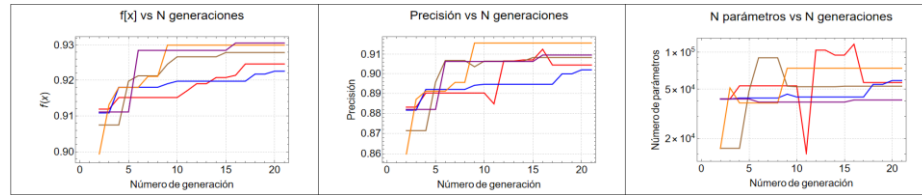


Fig. 3. Gráfico de la función de aptitud, precisión y número de parámetros para cinco ejecuciones del DeepGA en el conjunto de datos Fashion-MNIST.

Tabla 3. Estadísticos correspondientes a cinco ejecuciones del algoritmo de Deep-GA para el conjunto de datos Fashion-MNIST.

	Aptitud	Precisión	# de parámetros.
Mejor ejecución.	0.9305	0.9096	40898
Peor ejecución	0.9271	0.9020	58838
Promedio	0.9271	0.9080	56687.6
Desv. Estándar.	0.0034	0.0053	11929.7

Se observa mediante la Tabla 3 y Fig 3 que el valor de la función de aptitud para la mejor y peor ejecución difieren en 0.0034, de igual manera debido al valor *cercano* a cero de la desviación estándar se observa que las ejecuciones son robustas, por lo que el comportamiento denota una tendencia en los resultados a un valor de aptitud cercano a un máximo local, pues el valor de la función de aptitud no mejora significativamente en ninguna de las cinco ejecuciones. Lo anterior se ve reflejado en un error de clasificación promedio de 9.2% en la clasificación del conjunto Fashion-MNIST.

4.1 Comparación contra otras redes neuronales.

Una vez discutidos los resultados individuales para DeepGA es importante realizar la comparación con modelos del estado del arte: Resnet-18 [1], VGG-16 [3] y AlexNet [2]. Las comparaciones de DeepGA se realizan considerando la precisión y el número de parámetros, este proceso de comparación se hace con las bases de datos: MNIST y Fashion-MNIST, se usan como referencia los tres modelos y sus dos variantes: cuando la red está pre entrenada y cuando el proceso de entrenamiento lo realiza a partir de las correspondientes bases de datos.

El proceso comparativo se realiza buscando igualdad de condiciones, de manera que se llevan a cabo las evaluaciones con el mismo tamaño de batch en aquellos modelos en los cuales las capacidades técnicas lo permiten y el proceso de entrenamiento considera únicamente 10 épocas [10,11]. Un aspecto importante por destacar en la evaluación de las estructuras de las CNN es la realización de una interpolación para redimensionar las imágenes de entrada y sean congruentes con los requisitos de cada red.

Conjunto de datos MNIST:

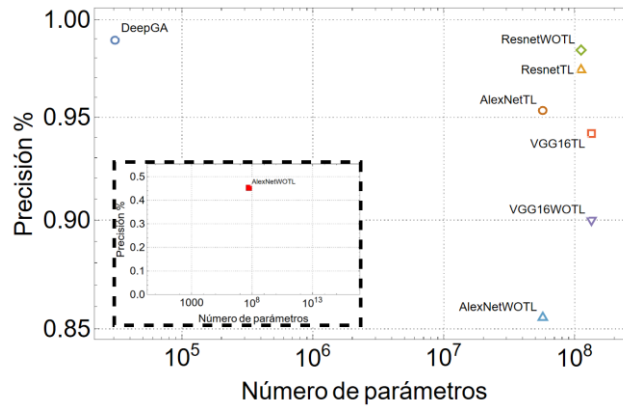


Fig. 4. Gráfico que muestra la precisión contra el número de parámetros del algoritmo DeepGA en la base de datos MNIST comparado modelos del estado del arte; Resnet-18, AlexNet, VGG-16. Las terminaciones *TL* y *WOTL* hacen referencia a que la ejecución se hizo con la red pre entrenada (Transfer Learning) o con la red sin entrenar (Without Transfer Learning).

Tabla 4. Estadísticos por modelo (MNIST). Las terminaciones *TL* y *WOTL* hacen referencia a que la ejecución se hizo con la red pre entrenada (Transfer Learning) o con la red sin entrenar (Without Transfer Learning). Se muestra el número de parámetros en millones.

Modelo	Precisión	# de parámetros. (Millones)
DeepGA.	0.9892	0.02410
Resnet-18 TL	0.9844	11.181
Resnet-18 WOTL	0.9863	11.181
AlexNet TL	0.9534	57.044
AlexNet WOTL	0.4551	57.044
VGG-16 TL	0.9419	134.301
VGG-16 WOTL	0.9000	134.301

En la Fig 4 y la Tabla 4 se observa que DeepGA genera redes neuronales con una mejor precisión y un número de parámetros considerablemente más pequeño que las redes con una estructura definida. Además, es posible determinar que DeepGA presenta una ventaja que oscila entre el 0.29 % comparado con el mejor modelo del estado del arte y un 53.41 % con el peor resultado de los modelos.

Conjunto de datos: Fashion-MNIST

Tabla 5. Estadísticos por modelo (Fashion-MNIST). Las terminaciones *TL* y *WOTL* hacen referencia a que la ejecución se hizo con la red pre entrenada (Transfer Learning) o con la red sin entrenar (Without Transfer Learning). Se muestra el número de parámetros en millones.

Modelo	Precisión	# de parámetros. (Millones)
DeepGA	0.9080	0.05668
Resnet-18 TL	0.7960	11.181
Resnet-18 WOTL	0.8054	11.181
AlexNet TL	0.8468	57.044
AlexNet WOTL	0.4869	57.044
VGG-16 TL	0.8504	134.301
VGG-16 WOTL	0.7371	134.301

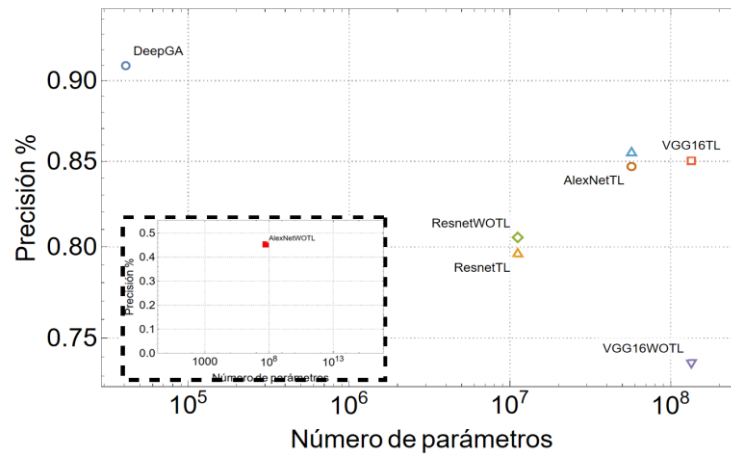


Fig. 5. Gráfico que muestra la precisión contra el número de parámetros del algoritmo DeepGA en la base de datos Fashion-MNIST comparado modelos del estado del arte; Resnet-18, AlexNet, VGG-16. Las terminaciones *TL* y *WOTL* hacen referencia a que la ejecución se hizo con la red pre entrenada (Transfer Learning) o con la red sin entrenar (Without Transfer Learning).

De igual manera que en el caso anterior, para el conjunto de Fashion-MNIST es posible advertir por medio de la Fig 5 y la Tabla 5 que DeepGA genera redes neuronales con una mejor precisión y un número de parámetros considerablemente más pequeño que las redes con una estructura definida. En particular para este conjunto de datos es de importancia notar que la diferencia en el porcentaje de precisión con los modelos del estado del arte varía entre 5.76 % y 42.11 % a favor de DeepGA.

5 Conclusiones

En este trabajo se presentó un análisis exploratorio del rendimiento de DeepGA analizando el comportamiento de la función de aptitud, precisión y número de parámetros por generación (véase Fig 2 y Fig 3) en donde se observa que para ambos conjuntos de datos el valor final de la función de aptitud varía muy poco en las 5 ejecuciones realizadas de manera que se observa cierta robustez. Sin embargo, pese a que en el caso del conjunto de datos de MNIST se obtiene un mejor valor para la función de aptitud (0.9305-0.9853), esto no sucede en el caso de Fashion-MNIST

donde se obtiene (0.9305-0.9271). Lo anterior puede sugerir que es necesario verificar las tendencias del comportamiento realizando más ejecuciones con un mayor número de generaciones para determinar si el algoritmo se estanca en algún máximo local.

En la comparación con estructuras del estado del arte, se observa a partir de Fig 4 y Fig 5 que los valores de precisión alcanzados por DeepGA superan en ambos casos a las redes con las que se le compara adquiriendo un porcentaje de (98.63±0.20 %) para el conjunto de datos MNIST y de (90.80±0.53%) para el conjunto Fashion-MNIST. De igual manera se muestra que el rendimiento de las redes es generalmente mejor cuando se hace la clasificación a partir del modelo pre entrenado, siendo Resnet-18 la única excepción. Se observa una ventaja en el porcentaje de clasificación en ambos conjuntos de datos mediante el uso de DeepGA que oscilan entre (0.29 % - 53.41 %) para MNIST y de (5.76 % - 42.11 %) para Fashion-MNIST.

Análisis complementarios al realizado en este trabajo y que pueden ayudar a la mejora de DeepGA son los siguientes: llevar a cabo la reparación de individuos analizando las capas convolucionales cuyas dimensiones no se ajusten a la salida de la capa previa y realizar la modificación de parámetros para el funcionamiento adecuado en conjuntos de datos cuyas imágenes sean de 3 canales (RGB).

Agradecimientos. Agradecemos al M.I.A Gustavo Vargas Hákim por sus comentarios acerca del funcionamiento del algoritmo DeepGA. Le agradecemos a los doctores Gabriel Acosta y Efrén Mézura por su disposición en el proceso de elaboración de este trabajo. A CONACYT por el apoyo mediante el programa de Becas Nacionales 2021.

Referencias

1. He, K., Zhang, X., Ren, S., & Sun, J. (2012). *Deep Residual Learning for Image Recognition*. Retrieved June 27, 2022, from <https://image-net.org/challenges/LSVRC/2015/>
2. Krizhevsky, A., & Inc, G. (2014). *One weird trick for parallelizing convolutional neural networks*.
3. Simonyan, K., & Zisserman, A. (2015). *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. <http://www.robots.ox.ac.uk/>
4. Verbancsics, P., & Harguess, J. (2015). Image classification using generative neuro evolution for deep learning. *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 488–493. <https://doi.org/10.1109/WACV.2015.71>
5. Wang, B., Sun, Y., Xue, B., & Zhang, M. (2019). A Hybrid GA-PSO Method for Evolving Architecture and Short Connections of Deep Convolutional Neural Networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11672 LNAI, 650–663. https://doi.org/10.1007/978-3-030-29894-4_52/COVER/
6. Vargas, G. A., Eng, H. B., Gabriel, H., & Mesa, A. (2021). *Neuroevolution of Convolutional Neural Networks for COVID-19 Classification in X-ray Images*.
7. Yann LeCun, Corinna Cortes, & Chris Burges. (n.d.). *MNIST handwritten digit database*. Retrieved June 28, 2022, from <http://yann.lecun.com/exdb/mnist/>
8. Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. <https://trends.google.com/trends/explore?date=all&q=mnist,CIFAR,ImageNet>
9. Vargas Hákim Gustavo Adolfo. (2021). *DeepGA-GitHub*. Retrieved June 28, 2022, from <https://github.com/GustavoVargasHakim/DeepGA>
10. Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. *IEEE Transactions on Cybernetics*, 50(9), 3840–3854. <https://doi.org/10.1109/TCYB.2020.2983860>
11. Gustavo Adolfo Vargas Hákim. (2021). *Neuroevolution of Convolutional Neural Networks for COVID-19 Classification in X-ray Images*