

Aplicación de algoritmos aproximados al diagnóstico/clasificación de enfermedades

Kevin Delsol, Miguel Isaia y Gabriela Minetti

Universidad Nacional de La Pampa
Facultad de Ingeniería

minettig@ing.unlpam.edu.ar

Resumen.

El diagnóstico de enfermedades puede formularse como un problema de clasificación, en consecuencia se trata de un problema NP-duro, como es el caso de las dos problemáticas que se pretenden resolver en este trabajo: clasificación, en benigno o maligno, de muestras de tumores de pacientes sospechados de sufrir de cáncer de mama; y clasificación, en negativo o positivo, de muestras de pacientes sospechados de padecer diabetes de tipo II. Por tal motivo, nuestra propuesta consiste en desarrollar algoritmos aproximados basados en perceptrones multicapa, en algoritmos genéticos y en algoritmos que hibridan estas opciones, para realizar diagnósticos confiables (clasificación) con respecto a estas enfermedades. Los experimentos numéricos permiten evaluar y comparar el rendimiento de las distintas propuestas utilizando conjuntos de datos reales. Los resultados muestran que nuestras propuestas logran resultados con errores de clasificación próximos a cero, además de, superar el desempeño de algoritmos propuestos en la literatura.

Keywords: Clasificación, diagnóstico médico, perceptrón multicapa, algoritmos genéticos, algoritmos híbridos.

1 Introducción

En la actualidad es posible modelar problemas complejos por medio de algoritmos aproximados, modelando el comportamiento de sistemas complejos, teniendo en cuenta las características particulares y factores aleatorios que los definen. Tal es el caso de los problemas pertenecientes al campo de la Biomedicina, en particular, el diagnóstico de enfermedades que puede formularse como un problema de clasificación, dado que es necesario examinar los datos médicos (análisis clínicos, examen físico, síntomas, resonancia, tomografía, etc.), o atributos, para determinar una enfermedad, o clase. Esto convierte al diagnóstico médico en un problema NP-duro, como es el caso de las dos problemáticas que se intentan solucionar en este trabajo: clasificación, en benigno o maligno, de muestras de tumores de pacientes sospechados de sufrir de cáncer de mama; y clasificación, en negativo o positivo, de muestras de pacientes sospechados de padecer diabetes de tipo II.

Las Redes Neuronales Artificiales (RNAs), las Metaheurísticas y los algoritmos híbridos se han convertido en herramientas eficientes para resolver esta clase de problemas [9, 10, 11, 12, 13, 14, 15]. Debido a la estructura masivamente paralela y distribuida, las RNAs tienen una gran capacidad para el aprendizaje, la generalización y la aproximación no lineal, y por lo tanto son adecuadas para la clasificación [1, 2]. Por su parte, las Metaheurísticas representan una familia de técnicas de optimización aproximada que mantiene el equilibrio entre la intensificación y la diversificación durante la búsqueda. Esto permite resolver eficientemente problemas de optimización complejos.

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

cientemente problemas complejos y difíciles, como es el caso de los problemas de clasificación [6]. También la hibridación de estas técnicas ha brindado soluciones de alta calidad a estos problemas [8].

En este trabajo, se pretende realizar un estudio sobre RNAs, Metaheurísticas y algoritmos híbridos, analizando características, ventajas y desventajas de su uso; para luego emplear y adaptar dichas técnicas a la clasificación y el diagnóstico de potenciales casos de cáncer de mama y diabetes tipo II. En particular se propone el diseño de dos RNAs basadas en el perceptrón multicapa, denominadas $\text{BackP}_{\text{base}}$ y $\text{BackP}_{\text{mod}}$, dos algoritmos metaheurísticos basados en algoritmos genéticos, GA_{1x} y GA_{2x} , y un algoritmo híbrido que combine estas técnicas, GA_{BackP} , con el objetivo de brindar diagnósticos confiables.

El artículo se organiza de la siguiente manera. En la Sección 2 se introducen los algoritmos que tratan los problemas en cuestión en la literatura, además se explican los aquí propuestos. En tanto que, en la sección 3 se especifica cómo se llevó a cabo la experimentación, el origen de los datos y, también, se analizan los resultados obtenidos. En la última sección se resumen nuestras conclusiones y se proyectan trabajos futuros.

2 Algoritmos aproximados empleados en problemas de clasificación

En esta sección se describen algoritmos aproximados que resuelven problemas de clasificación (por ej. diagnóstico de enfermedades), tales como: perceptrón multicapa, algoritmos genéticos e híbridos. Además, cada una de estas descripciones irá acompañada por la explicación de nuestra propuesta algorítmica.

2.1 Perceptrón multicapa (*Multi-Layer Perceptron*, MLP)

Un MLP es una red *feed-forward* con aprendizaje supervisado [3, 4]. El tipo de aprendizaje más utilizado por un perceptrón es el método *backpropagation* (propagación del error hacia atrás) basado en la generalización de la regla Delta [4]. A pesar de sus limitaciones, este método ha contribuido a ampliar de forma considerable el rango de aplicaciones de los MLPs (diagnóstico de enfermedades, clasificación de señales, simulación de la dinámica del brazo de un robot, puntuación para la solicitud de créditos, etc.)

En este tipo de aprendizaje, primero se aplica un patrón de entrada como estímulo a la primera capa de las neuronas de la red, el cual se propaga a través de todas las capas ocultas hasta generar una salida (paso *forward*). Dicha propagación se lleva a cabo al aplicar una función de activación a la suma ponderada, que surge de la combinación lineal de las entradas y las conexiones (o pesos) sinápticos correspondientes. Luego se evalúa el error cometido al comparar el resultado obtenido por las neuronas de salida con el patrón de salida, por medio de una función de costo. A continuación, estos errores se transmiten hacia atrás (paso *backward*), partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aporte relativo al error total. En base a esto, se reajustan los pesos de conexión de cada neurona, de manera que la próxima vez que se presente el mismo patrón, la salida esté más cercana a la deseada.

Para acelerar el proceso de aprendizaje se puede añadir un término denominado momento o *momentum* [4], el cual acelera la convergencia del algoritmo. Otra forma de mejorar el proceso de aprendizaje es adaptar la tasa de aprendizaje, η , durante dicho proceso, conocido como η adaptativo [4], y consiste en modificar el valor de η durante el entrenamiento con el fin de evitar que el error se estanque debido a la presencia de óptimos locales.

Nuestra propuesta consiste en desarrollar dos MLPs, denominados $\text{BackP}_{\text{base}}$ y $\text{BackP}_{\text{mod}}$. $\text{BackP}_{\text{base}}$, que usan el aprendizaje *backpropagation*. En tanto que, $\text{BackP}_{\text{mod}}$ modifica a $\text{BackP}_{\text{base}}$ al implementar las técnicas de *momentum* y η adaptativo, con el fin de regular la velocidad de aprendizaje y por ende la convergencia del algoritmo. Ambos algoritmos se implementaron siguiendo las pautas especificadas por Hertz et. al en [4].

La fig. 1 representa la arquitectura de $\text{BackP}_{\text{base}}$ y de $\text{BackP}_{\text{mod}}$, donde n representa el número de neuronas de entrada y m refleja la cantidad de neuronas en la capa oculta. Las conexiones sinápticas están representadas por las matrices: \mathbf{w}^1_{ij} (con $i = 1, \dots, m$ y $j = 1, \dots, n$) y \mathbf{w}^2_{ij} (con $j = 1, \dots, m$), según se muestra en la fig. 2.

Si bien, en ambos casos se utiliza una capa de entrada, una oculta y una de salida, la diferencia se presenta en el número de neuronas de entrada, dada la cantidad de atributos de cada caso de estudio (8 para cáncer de mama y 9 para diabetes). La capa oculta consta de tres neuronas, número establecido por medio de un proceso de configuración que se explica en la Sección 4. La capa de salida se compone de una única neurona, que representa la decisión de clasificación tomada por la red para cada caso en particular.

Para ambos MLPs \mathbf{w}^1_{ij} y \mathbf{w}^2_{ij} se crean con números aleatorios uniformemente distribuidos, en el rango $[-0.5, 0.5]$, y es necesario configurar los parámetros β , η , error medio de aprendizaje deseado y número de épocas. El parámetro β es una constante utilizada para el cálculo de la función de activación tangente hiperbólica, η por su parte, es la tasa de aprendizaje utilizada para la corrección de los pesos sinápticos durante el paso hacia atrás. Las funciones de activación evaluadas para su uso son: tangente hiperbólica, exponencial y lineal. En cuanto a las funciones de costo se consideran el error cuadrático medio y la medida entrópica [4].

Los patrones de entrada-salida seleccionados para entrenar a la red corresponden al 70% de los datos de cada caso de estudio, el resto es utilizado en la etapa de testeo. Además, es necesario normalizar los patrones para establecer una comparación válida con los resultados de las funciones de activación tangente hiperbólica y exponencial. Por último y con el objetivo de evitar que la red memorice durante el entrenamiento, ambas propuestas utilizan una actualización incremental [4].

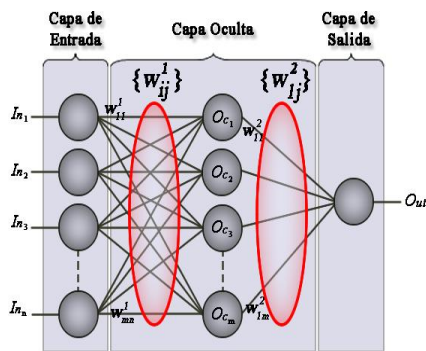


Fig. 1. Arquitectura general de los MLPs.

w^1_{11}	w^1_{12}	\dots	w^1_{1n}	} w^1_{ij}
w^1_{21}	w^1_{22}	\dots	w^1_{2n}	
\vdots	\vdots	\ddots	\vdots	
w^1_{m1}	w^1_{m2}	\dots	w^1_{mn}	
w^2_{11}	w^2_{12}	\dots	w^2_{1m}	} w^2_{ij}

Fig. 2. Matrices de pesos sinápticos.

En particular para $\text{BackP}_{\text{mod}}$, se agregan las constantes α , a , b y *pasos*. Tanto α , a y b son utilizadas para mejorar η en las técnicas de *momentum* y η adaptativo; mientras que *pasos* es el número de iteraciones tras el cual se verifica el valor de η .

2.2 Algoritmos Genéticos (*Genetic Algorithms, GAs*)

Los GAs [7] son un tipo popular de Metaheurísticas, en especial de algoritmos evolutivos, que simula la evolución de individuos por medio del proceso de selección natural Darwiniano. Computacionalmente hablando, esto se refiere a que realizan una búsqueda multidireccional manteniendo una población de potenciales soluciones (individuo) y fomentando, por un lado, la formación de información (por medio de la mutación), y por otro lado, el intercambio de información (por medio del *crossover*) entre dichas soluciones.

La selección de los individuos a reproducirse (*crossover* y mutación), se realiza a través de operadores de selección probabilísticos [6] (ruleta, ranking, torneo binario, entre otros).

Generación tras generación, los individuos de la población actual son sustituidos por los recién creados. Este reemplazo usualmente intenta mantener los mejores individuos (elitismo) eliminando los peores. Dependiendo de si se tiene en cuenta la antigua población se pueden obtener dos tipos de estrategias de reemplazo: $(\mu; \lambda)$ si el reemplazo se realiza utilizando únicamente a la población de hijos, y $(\mu + \lambda)$ si el reemplazo se realiza seleccionando μ individuos de la unión de la población antigua y la de hijos [6].

Los dos GAs propuestos, GA_{1X} y GA_{2X} , utilizan números reales en la representación de cada individuo, la selección de los padres se realiza por medio de un torneo binario, método de selección que es utilizado también en el proceso de reemplazo $(\mu + \lambda)$ con elitismo. Además aplican el método *swap* de un gen (elemento de la solución) elegido al azar para llevar a cabo la mutación de un individuo bajo una determinada probabilidad (p_m).

La diferencia entre GA_{1X} y GA_{2X} radica en el método de *crossover* empleado. Por un lado, GA_{1X} utiliza *crossover* de un punto, con el fin de testear un GA estándar. En tanto que, GA_{2X} aplica el *crossover* de dos puntos, cuyo objetivo es introducir mayor variación en cada descendencia generada. Ambos operadores se aplican bajo una cierta probabilidad (p_c).

La fig. 3 ilustra el origen de los genes de cada individuo; donde cada uno se compone de los pesos sinápticos de un MLP. Los individuos se estructuran en forma de vector, a diferencia de la estructura matricial utilizada en los MLPs. En otras palabras, cada peso de conexión representa un gen del individuo, siendo creados con valores aleatorios en el rango [-0.5, 0.5] siguiendo una distribución uniforme.

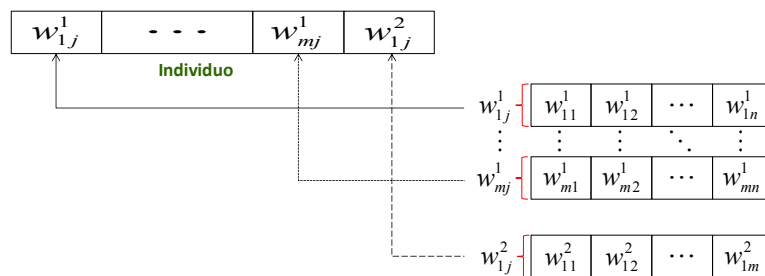


Fig. 3. Composición de cada individuo de la población de un GA.

El *fitness* mide la capacidad del individuo para resolver los problemas planteados, es decir, el promedio de error cometido al intentar clasificar patrones de cáncer de mama o diabetes. Mientras menor sea dicho error, el *fitness* del individuo será mejor. En otras palabras, la evaluación del *fitness* de cada individuo se corresponde con el cálculo del error medio obtenido al ejecutar sólo el paso hacia adelante del MLP durante una época.

2.3 Metaheurísticas Híbridas

Combinaciones de algoritmos tales como metaheurísticas, programación matemática, programación con restricciones y técnicas de aprendizaje automático (*machine learning*) proveen algoritmos de búsqueda muy poderosos para la resolución de problemas NP-duros [6, 8]. Nuestra propuesta híbrida, GA_{BackP} , combina metaheurísticas con técnicas de aprendizaje automático, al integrar la facilidad de adaptación de los GAs con el aprendizaje automático presente en los MLPs. El GA brinda la estructura de contención, y el MLP es usado como el operador de mutación.

Este nuevo operador de mutación, a diferencia de los implementados por los GAs, no aplica una mutación “a ciegas”, sino una mutación “inteligente”. Esto se debe que el MLP muta al individuo intentado minimizar el error de clasificación del mismo. Con esto se espera que el individuo aumente su probabilidad de supervivencia durante la evolución.

La fig. 4 muestra el proceso de mutación de GA_{BackP} , que se logra ingresando al individuo en la red, y entrenando la misma con el fin de refinar el *fitness* de dicho individuo. En primer lugar, el individuo es descompuesto en dos vectores: w^{1}_{ij} y w^{2}_{1j} . Los vectores se reestructuran en las matrices que conforman las conexiones sinápticas de la red, luego se realiza un entrenamiento del MLP según se explicó en el diseño de los mismos. Por último se lleva a cabo el trabajo inverso a la descomposición del individuo, convirtiendo a las matrices en vectores y obteniendo un individuo mutado.

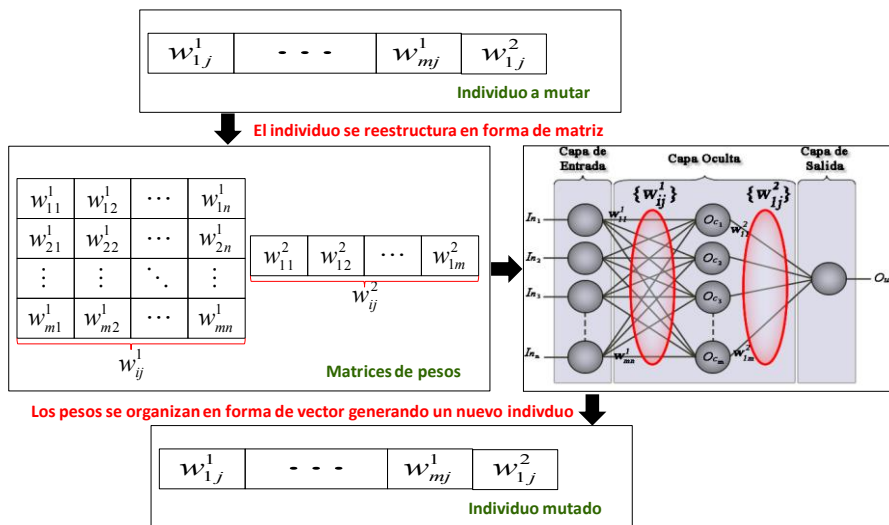


Fig. 4. Mutación de un individuo por medio de la RNA.

3 Experimentación y Análisis

En este apartado, se explica la metodología utilizada para llevar a cabo las experimentaciones, se analizan los resultados obtenidos y comparan los mismos con los publicados por Alba y Chicano en [9]. En los experimentos se usan datos de casos reales almacenados en repositorios de uso público, y se realizan sobre PCs idénticas equipadas con: procesador Intel Atom CPU N2600, 1.6GHz, memoria RAM de 2GB DDR3 y Microsoft Windows 7 Ultimate x86.

En el caso del diagnóstico de cáncer de mama, los datos utilizados pertenecen a la Universidad de Wisconsin¹ y constan de 699 instancias compuestas de 8 atributos cada uno. El 65, 5% (458) de estos casos corresponden a instancias clasificadas como benignas y el resto como malignas. En tanto que, los datos utilizados para el diagnóstico de diabetes pertenecen al National Institute of Diabetes and Digestive and Kidney Diseases² y corresponden a pacientes pertenecientes al grupo aborigen Pima. Los mismos consisten de 768 instancias con 9 atributos cada uno, de las cuales 500 (65,1%) son clasificadas como negativas, es decir, pacientes no diabéticos y 268 (34.9%) son positivas, es decir corresponden a pacientes diabéticos.

Tabla 1. Configuración paramétrica para cada algoritmo y caso de estudio.

Alg.	BackP _{base}	BackP _{mod}	GA _{1X}	GA _{2X}	GA _{BackP}	BackP _{base}	BackP _{mod}	GA _{1X}	GA _{2X}	GA _{BackP}
Caso de estudio	Cáncer de mama					Diabetes Tipo II				
Función de Act.	TanH					TanH				
Función de Costo	Error Cuadrático Medio					Error Cuadrático Medio				
β	0,5					0,5				
η	0,5					0.01				
n y m	8 y 3					9 y 50				
Épocas de entrenam.	10000	10000	-----		15	10000	10000	-----		20
Pasos		15					15			
α	-----	0.01			-----	-----	0.001			-----
A		0.001					0.05			
B		0.05					0.01			
Tamaño poblacional					128					128
p_c					0.99					0.99
p_m					0.2					0.2
Total de generaciones					500					500
					500					500
					250					250

La configuración paramétrica usada en los experimentos (ver tabla 1) surge de una comparación de los resultados obtenidos al darle a cada parámetro diferentes valores y realizando todas las posibles combinaciones de los mismos³. Para cada algoritmo y caso de estudio, se elige la configuración de parámetros que mejores resultados ofrecen, en términos de errores medios de entrenamiento. Con las configuraciones elegidas, cada algoritmo se ejecuta 30 veces para obtener una muestra confiable y poder realizar un análisis estadístico de los resultados, con el fin de determinar si existen diferencias significativas entre los mismos y cuáles son los que difieren. Para esto último se aplica la prueba de ANOVA si los datos

¹ <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data>

² <http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data>

³ Ver apéndice 2 en [8].

sigan una distribución normal, o el test no paramétrico Kruskal Wallis en caso contrario; ambos con un nivel de confianza del 95%.

El análisis de los resultados obtenidos en la experimentación se desprende de los diagramas de box-plot mostrados en las figuras 5 y 6. Estos gráficos permiten visualizar el error cometido por cada algoritmo en todas las ejecuciones, y reflejan los resultados de los análisis estadísticos. Es decir, si existen o no diferencias significativas en el comportamiento de los mismos.

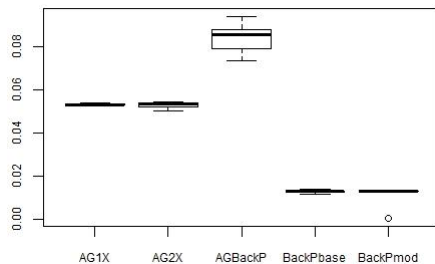


Fig. 5. Diagrama de box-plot del error cuadrático medio obtenido por cada algoritmo para el caso de estudio: Cáncer de mama.

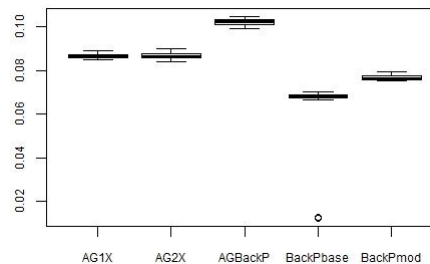


Fig. 6. Diagrama de box-plot del error cuadrático medio obtenido por cada algoritmo para el caso de estudio: Diabetes tipo II.

Por un lado analizamos los resultados relacionados con el caso de estudio ‘Cáncer de mama’ mostrados en la fig. 5. En primer lugar, se observa que, los algoritmos que clasifican con errores próximos a cero son $BackP_{base}$ y $BackP_{mod}$. Además, no existen diferencias estadísticas en sus comportamientos, siendo $BackP_{base}$ quien obtiene el mínimo error. En tanto que, GA_{1X} y GA_{2X} clasifican con errores relativamente bajos (aproximadamente del 5%) y presentan comportamientos estadísticamente similares, siendo GA_{2X} quien comete el menor error. Por último, y con errores que varían entre el 5 y 7% se encuentra GA_{BackP} , cuyo comportamiento es significativamente distinto al de los otros 4 algoritmos.

Por otra parte, la fig. 6 permite analizar los resultados obtenidos para el caso de estudio ‘Diabetes tipo II’. Los cinco algoritmos se comportan de forma similar que para el caso de cáncer de mama, pero esta vez el error mínimo es mayor al 6%.

En [9], los autores proponen un GA, un MLP denominado BP y dos algoritmos que hibridan estas técnicas bajo el nombre GABP y GALM para resolver los dos casos de estudio tratados aquí. La métrica usada para la comparación es conocida como Porcentaje de Errores de Clasificación (PEC), que indica el porcentaje clasificado erróneamente por el algoritmo en cuestión (vec. 1). El promedio de estos valores para $BackP_{base}$, GA_{2X} , GA_{BackP} , BP[9], GA[9], GABP[9] y GALM[9], en cada caso de estudio, se muestran en la tabla 2.

$$PEC = ErrorMedio * 100 \quad (1)$$

Al analizar la tabla 2 se observa que, en ambos casos de estudio, $BackP_{base}$ y GA_{2X} incurren en errores de clasificación menores a los cometidos por BP y GA, respectivamente. Esta disminución de los valores de PEC varía entre un 38 y un 77%. No obstante, en los dos trabajos, se advierte que los algoritmos basados en MLP mejoran los resultados obtenidos por los basados en genéticos entre un 20 y un 94%.

Table 2. Promedio de los valores de PEC por cada algoritmo y caso de estudio.

Caso de estudio /Alg.	$BackP_{base}$	BP	GA_{2X}	GA	GA_{BackP}	GABP	GALM
Cáncer de mamas	0.56	0.91	5.07	16.76	7.97	1.43	0.02
Diabetes tipo II	6.52	21.76	8.2	36.46	10.1	36.46	28.29

En cuanto a los algoritmos híbridos, GA_{BackP} , GABP y GALM, se observan diferencias significativas en el comportamiento según el caso de estudio analizado. Esto es, para el cáncer de mama, los valores de PEC correspondientes a las dos versiones híbridas propuestas en [9] son entre 5 y 300 veces menor a los de GA_{BackP} . En cambio, para el segundo caso es GA_{BackP} quien incurre en un error 2 y 3 veces más pequeño. Sin embargo, estos híbridos no superan el desempeño de los MLPs.

4 Conclusiones

En este artículo se desarrollaron dos perceptrones multicapa, dos algoritmos genéticos y un algoritmo que hibrida los anteriores para resolver problemas de clasificación pertenecientes al área de la medicina: diagnóstico de casos de cáncer de mama y diabetes tipo II. Estos algoritmos fueron evaluados usando casos de prueba reales publicados por instituciones reconocidas por la comunidad científica y su comportamiento fue comparado con el de otros algoritmos publicados en la literatura.

Realizadas las experimentaciones, se observó que los algoritmos basados en MLPs son los que mejores resultados ofrecen, seguidos por los GAs, dejando al algoritmo híbrido en último lugar. De la contrastación con otros algoritmos de la literatura surgió que en cinco de las seis comparaciones realizadas, los algoritmos aquí propuestos obtienen mejores resultados.

Como futuras líneas de investigación, en primer lugar se intentará mejorar el desempeño de los GAs incorporando heurísticas en los operadores genéticos. Por último, y con el objetivo de que el algoritmo híbrido alcance los resultados esperados se analizarán mejoras en la implementación del operador de mutación para acelerar la convergencia al error mínimo.

Referencias

1. J. T. Palma Méndez, R. Marín Morales, "Inteligencia artificial: Técnicas, métodos y aplicaciones", McGraw-Hill, Madrid, España, 2008.
2. C. T. Leonides, "Neural Network Systems Techniques and Applications", Vol. 1 a 7, Academic Press, San Diego, CA, USA, 1998.
3. J. R. Hiler, V. J. Martínez, "Redes Neuronales Artificiales", Alfaomega-Rama, México DF, 2000.
4. J. Hertz, R. G. Palmer, A. S. Krogh. "Introduction to the Theory of Neural Computation", Perseus Publishing, 1991.
5. K. Delsol, M. Isaia, "Utilización de Redes Neuronales Artificiales y Algoritmos Genéticos para diagnosticar enfermedades", Facultad de Ingeniería, Universidad Nacional de La Pampa, 2015 .
6. E.-G. Talbi, "Metaheuristics: From Design to Implementation", Wiley & Sons, USA, 2009.
7. J. Holland, "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence", Oxford, UM, England, 1975.
8. E.-G. Talbi, "A taxonomy of hybrid metaheuristics." Journal of Heuristics, 2002.
9. E. Alba, J. F. Chicano, "Training Neural Networks with GA Hybrid Algorithms", Genetic and Evolutionary Computation (GECCO), ISBN 978-3-540-22344-3, pág. 852-863 , 2004.
10. A. Bacha, K. Kalti, N.E. Ben Amara, B. Solaiman, "Microcalcifications detection in mammograms based on Ant Colony Optimization and Markov Random Field", 6th International Conference on Soft Computing and Pattern Recognition, SoCPaR 2014, pp. 191-196, 2015.
11. S.K. Suguna, S.U. Maheswari, "Classification of feature extracted, selected and segmented mammogram image using hybrid algorithm-monkey search optimization(MSO) and support vector machine(SVM)", Research Journal of Applied Sciences, 9 (2), pp. 110-118, 2014.
12. A.N. Machraoui, M.A. Cherni, M. Sayadi, "Ant Colony optimization algorithm for breast cancer cells classification", 2013 International Conference on Electrical Engineering and Software Applications, 2013.
13. A. Askarzadeh, A. Rezaezadeh, "Artificial neural network training using a new efficient optimization algorithm", Applied Soft Computing Journal, 13 (2), pp. 1206-1213, 2013.
14. M.S. Rauner, W.J. Gutjahr, K. Heidenberger, J. Wagner, J. Pasia, "Dynamic policy modeling for chronic diseases: Metaheuristic-based identification of pareto-optimal?", Operations Research, 58 (5), pp. 1269-1286, 2010.
15. N. Belacel, M. Čuperlović-Culf, M. Laflamme, R. Ouellette, "Fuzzy J-Means and VNS methods for clustering genes from microarray data", (2004) Bioinformatics, 20 (11), pp. 1690-1701.