

**Diagnóstico de retinopatía diabética utilizando algoritmos de inteligencia artificial**

**Héctor Fabio Ocampo Arbeláez**

**Universidad del Valle Sede Tuluá  
Facultad de ingeniería  
Escuela de ingeniería de sistemas y computación  
Tuluá Valle del Cauca  
2017**

# 1. Estado del arte

Tabla 1: Estado del arte

Nombre del artículo	Algoritmos Utilizados	Resultados
Detección y clasificación de Retinopatía Diabética no Proliferativa usando una Red Neuronal de Retropropagación [1].	<ul style="list-style-type: none"><li>■ Redes neuronales.</li></ul>	Tuvo una precisión del 95 % en su diagnóstico.
An ensemble-based system for automatic screening of diabetic retinopathy [2].	<ul style="list-style-type: none"><li>■ Redes neuronales.</li><li>■ SVM.</li><li>■ Random Forest.</li><li>■ Árboles de decisión.</li></ul>	Usó diferentes algoritmo de aprendizaje, pero el que mejor resultado dio, fue SVM con una precisión mayor al 95 % (Citado por 35).
Machine Learning on the Diabetic Retinopathy Debrecen Data Set[3]	<ul style="list-style-type: none"><li>■ Redes neuronales.</li><li>■ SVM.</li><li>■ Random Forest.</li><li>■ K-NN.</li></ul>	SVM tuvo la mejor precisión con un 76 %

Nombre del artículo	Algoritmos Utilizados	Resultados
An Improved Medical Decision Support System to Identify the Diabetic Retinopathy Using Fundus Images[4].	<ul style="list-style-type: none"> <li>▪ SVM.</li> </ul>	Tiene una precisión del 97 %, a diferencias del anterior. Este usa como entreda directamente las imágenes de la retina. (Citado por 6)
An Expert Support System for Breast Cancer Diagnosis using Color Wavelet Features[5].	<ul style="list-style-type: none"> <li>▪ SVM.</li> <li>▪ Redes bayesianas.</li> <li>▪ Redes Neuronales.</li> </ul>	SVM tuvo la mejor precisión con un 98 % (Citado por 16).
Tuberculosis Disease Diagnosis Using Artificial Neural Network Trained with Genetic Algorithm[6].	<ul style="list-style-type: none"> <li>▪ Redes neuronales entrenadas por algoritmos genéticos.</li> </ul>	Tuvo una precisión del 94 % por ciento, mejorando en un 2 % otros estudios anteriores con redes neuronales. (Citado por 44)

Nombre del artículo	Algoritmos Utilizados	Resultados
Un sistema experto difuso en la Web para diagnóstico de diabetes[7].	<ul style="list-style-type: none"> <li>■ Lógica difusa.</li> </ul>	Fue probado por varios médicos, pacientes con y sin diabetes, mostrando que ayuda muchos a los médicos a dar un diagnóstico más certero, teniendo en cuenta más variables y ser más óptimos con el tiempo. (Citado por 1)
Artificial neural networks applied to cancer detection in a breast screening programme[8].	<ul style="list-style-type: none"> <li>■ Redes Neuronales.</li> </ul>	Usó 9 modelos de redes neuronales, siendo el primero con 32 neuronas y aumentando hasta el modelo 9 con 484. después del entrenamiento y las pruebas, se encontró que los modelos 5, 6 y 7 tenían los mejores resultados, y siendo el modelo 5 con 144 neuronas la que mejor precisión tuvo, con un 75 %. (citado por 27)
A Clinical Decision Support System with an Integrated EMR for Diagnosis of Peripheral Neuropathy[9].	<ul style="list-style-type: none"> <li>■ Lógica difusa.</li> </ul>	El sistema puede diagnosticar 4 diferentes neuropatías y fue testeado con ayuda de expertos, presentando una precisión del 93 %. (Citado por 4)
A Study on Hepatitis Disease Diagnosis Using Probabilistic Neural Network[10].	<ul style="list-style-type: none"> <li>■ Redes neuronales Probabilísticas(PNN).</li> </ul>	PNN tiene una precisión de 91,2 %, una precisión mayor que muchos otros modelos de redes neuronales para esta enfermedad (Citado por 25).

Nombre del artículo	Algoritmos Utilizados	Resultados
A Study on Hepatitis Disease Diagnosis Using Multilayer Neural Network with Levenberg Marquardt Training Algorithm[11].	<ul style="list-style-type: none"> <li>■ Redes neuronales Entrenadas por Levenberg Marquardt(MLNN).</li> </ul>	MLNN tiene una precisión de 91,87 %, siendo así una red neuronal con mayor precisión, esto puede ser a que LM le proporciona una mayor velocidad de convergencia (Citado por 45).
Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ANN, SVM, and classifier fusion[12].	<ul style="list-style-type: none"> <li>■ Lógica difusa.</li> <li>■ ANN.</li> <li>■ SVM.</li> <li>■ Clasificador de fusión.</li> </ul>	Se usó el sistema en dos bases de datos diferentes, en los dos casos las redes neuronales tuvo la mejor precisión con un 94 %, seguida por la lógica difusa con un 93 % de precisión (Citado por 4).
Vessel segmentation and microaneurysm detection using discriminative dictionary learning and sparse representation[13].	<ul style="list-style-type: none"> <li>■ Discriminative dictionary learning.</li> </ul>	Detecta los microaneurismas y vasos de la retina del ojo, es usado en dos bases de datos diferentes, y tiene una precisión del 94 % (Citado por 5).

## 2. Algoritmos Usados

Las técnicas más usadas en la tabla anterior son:

- Redes neuronales: usada 8 veces.
- SVM: usada 5 veces.

### 2.1. Redes neuronales artificiales

Esta técnica está motivada en modelar la forma del procesamiento de la información en sistemas nerviosos biológicos, principalmente por la forma del funcionamiento del cerebro humano. Dado esto una red neuronal artificial es un procesador de información construido por unidades sencillas de procesamiento llamadas neuronas, algunas características son: adquirir conocimiento a partir de la experiencia, adaptabilidad, tolerancia a fallas y comportamiento no lineal [14].

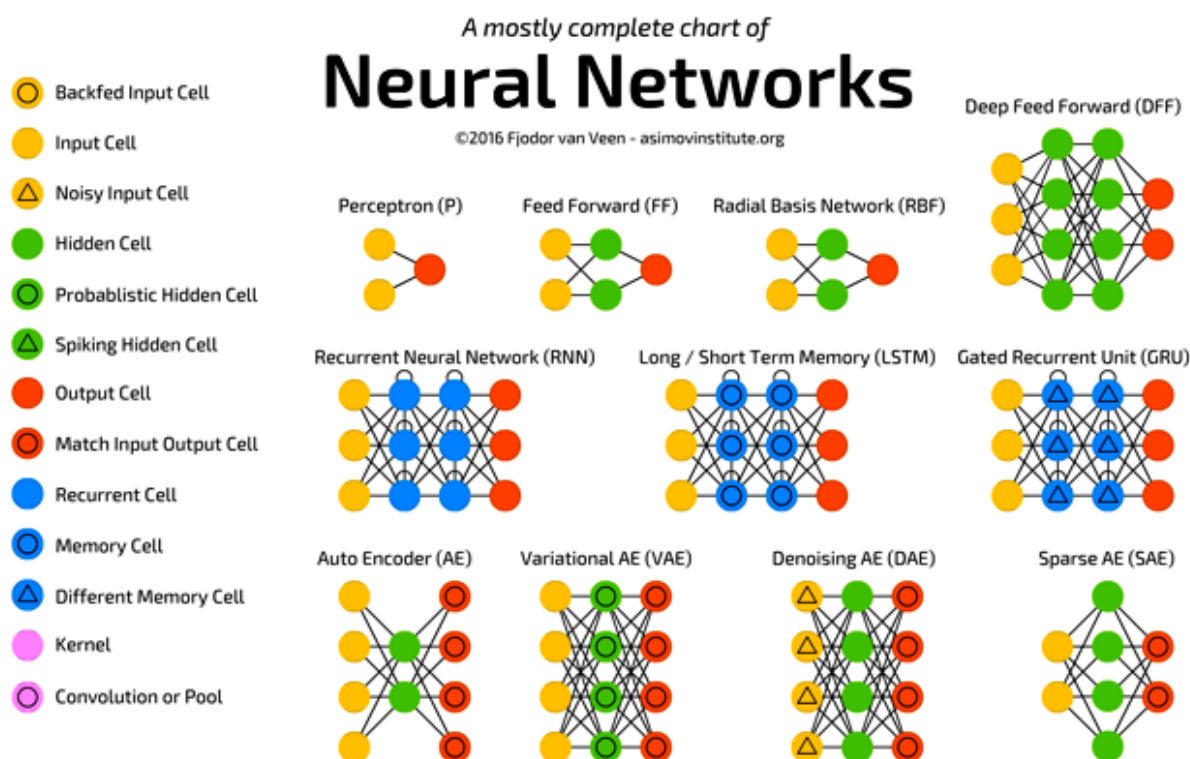


Figura 1: Diferentes arquitecturas de redes neuronales.

1

La figura 1 muestra diferentes arquitecturas de redes neuronales, donde cada una de ellas puede tener un mejor comportamiento sobre un problema en específico, y a medida

<sup>1</sup>Imagen tomada de: [http://www.example.com/the\\_image.png](http://www.example.com/the_image.png) ultima visita: 02-10-2017

que avanza los estudios de la inteligencia artificial nuevas arquitecturas de redes neuronales aparecen para ampliar y mejorar esta área.

La construcción de una red neuronal consiste en generar un modelo o seleccionar y adaptar uno ya existente, identificar las entradas, salidas y calcular los pesos de la sinapsis. Este último es importante ya que de estos pesos depende la decisión final que tendrá la red neuronal. Para calcular estos pesos existen varias técnicas que consisten en prueba y error hasta llegar a nivel de satisfactibilidad, a este proceso se le llama aprendizaje de la red neuronal.

## 2.2. Máquinas de soporte vectorial

Son un conjunto de algoritmos centrados principalmente en la clasificación y regresiones lineales, se encuentra entre los algoritmos de aprendizaje supervisado, para la clasificación crea un conjunto de hiperplanos en espacios dimensionales altos [15].

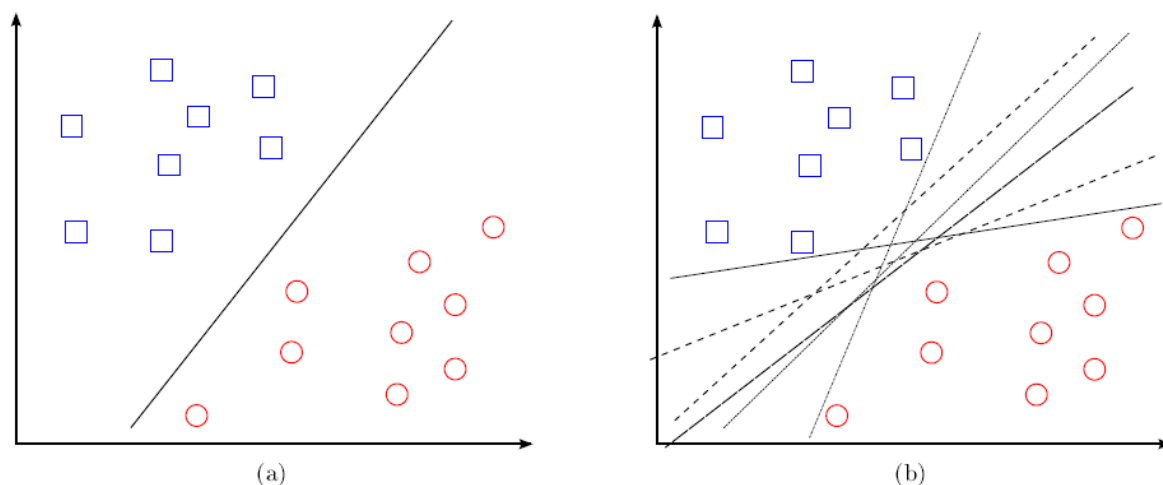


Figura 2: Hiperplanos de separación.

2

La figura 2 muestra dos gráficas donde (a) solo contiene un hiperplano de separación, (b) muestra más hiperplanos de separación de los datos. Y aunque se puedan crear diferentes hiperplanos de separación se debe de encontrar un hiperplano óptimo el cual esté equidistante de las dos categorías de datos.

## 3. Selección

La técnica será seleccionada a partir de lo siguiente:

---

<sup>2</sup>imagen tomada de: [urlhttp://www.ia.uned.es/ejcarmona/publicaciones/\[2013-Carmona\]%20SVM.pdf](http://www.ia.uned.es/ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf) última visita 02-10-2017.

- Número de implementaciones en la tabla de antecedentes.
- Citaciones de los artículos en la tabla de antecedentes.
- Mejores Resultados obtenidos en la tabla de antecedentes.

Dado esto se concluye que la técnica seleccionada será Redes Neuronales:

- Redes neuronales tiene el mayor número de implementaciones en la tabla de antecedentes, con 8 implementaciones a comparación de SVM con 5 veces.
- Redes neuronales tiene el artículo con mayor número de citaciones, «Redes neuronales Entrenadas por Levenberg Marquardt (MLNN) [11]» con 45 citaciones, el segundo artículo en número de citaciones también implementa redes neuronales, «Tuberculosis Disease Diagnosis Using Artificial Neural Network Trained with Genetic Algorithm [6]» con 44 citaciones, y el tercer artículo en número de citaciones implementa diferentes algoritmos entre ellos redes neuronales y SVM, «An ensemble-based system for automatic screening of diabetic retinopathy [2]», con 35 citaciones.
- Redes neuronales junto con SVM tienen porcentaje de precisión muy alto, teniendo a SVM con su mayor precisión en un 98 % y a redes neuronales con su mejor precisión en un 95 %.

## 4. Implementación

Dada la técnica seleccionada que fue Redes Neuronales, se hará la implementación de ésta. Redes Neuronales puede tener muchas variaciones ya sea en su arquitectura, tamaño o algoritmo de entrenamiento.

para el actual proyecto se implementará redes neuronales con arquitectura perceptrón entrenadas por dos algoritmos diferentes:

### 4.1. Backpropagation:

Backpropagation o Propagación hacia atrás, es un algoritmo de entrenamiento de Redes Neuronales, este algoritmo utiliza los patrones de salida de la red y la salida obtenida para calcular el error de la salida [14].

Este error se propaga desde la capa de salida, hasta la capa de entrada, hacia atrás. afectando los pesos de cada neurona dado un factor de entrenando y su contribución con la salida, así las neuronas que son más determinantes tendrán más cambios a comparación con las neuronas que no son tan significativas.

Para el actual proyecto se implemento el algoritmo backpropagation en una red neuronal de una capa oculta, y se verá de la siguiente forma:



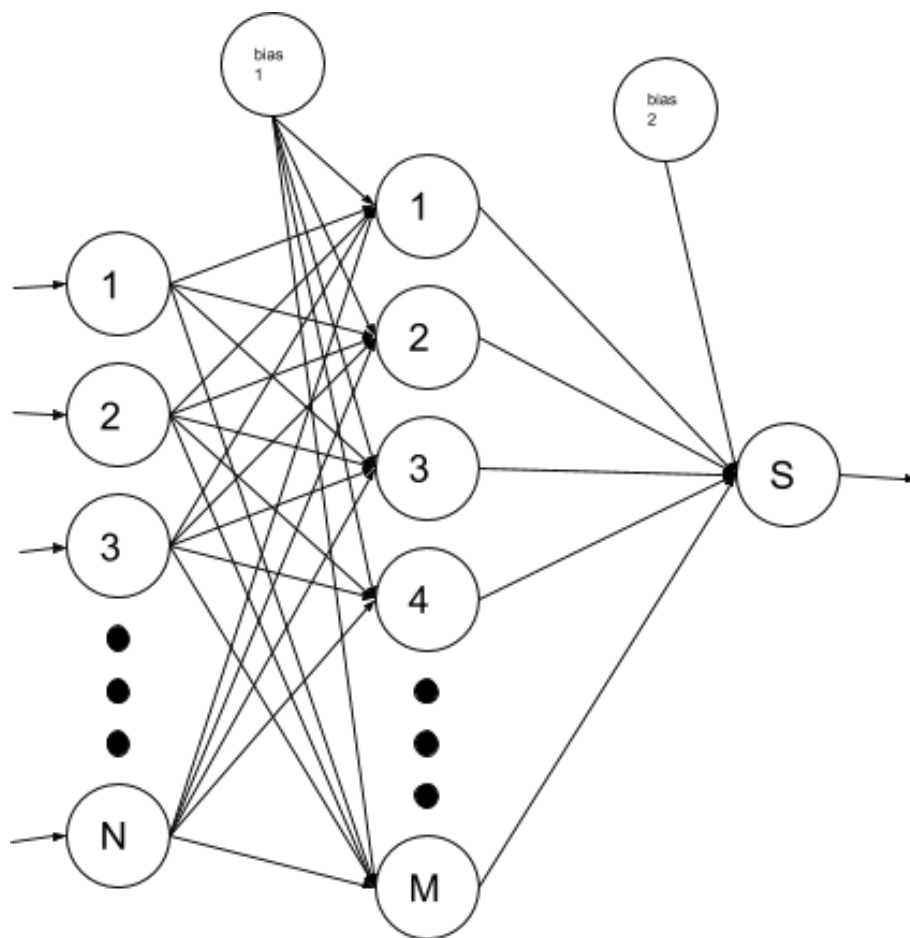


Figura 3: Arquitecturad de la Red Neuronal.

## 4.2. Algoritmos Genéticos:

Los algoritmo genéticos es una técnica computaciones bioinspirada de la evolución de Darwin. ésta es una técnica exploratoria muy útil en problemas de optimización [16]. como lo es el problema de optimizar los pesos de la red neuronal.

La implementación del algoritmo genético puede variar dado diferentes factores, como lo son población inicial, forma de reproducción, porcentaje de mutación, fórmula de mutación, selección, entre otras. todas estas variables pueden generar una gran catidad de posibilidades

### 4.2.1. Cromosoma:

Para la implementación del algoritmo evolutivo es importante determinar el genotipo y fenotipo del cromosoma, y cómo se representará computacionalmente para su ejecución.

El **Fenotipo:** será la repreceñación de las matrices de pesos y bias usados en la Red neuronal de la Figura 3

$$PesosCapaOculta_{(M*N)} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} & \dots & W_{1N} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} & \dots & W_{2N} \\ \vdots & & & & & & \\ \vdots & & & & & & \\ W_{M1} & W_{M2} & W_{M3} & W_{M4} & W_{M5} & \dots & W_{MN} \end{bmatrix}$$

$$PesosCapaSalida_{(1*M)} = [P_{11} \ P_{12} \ P_{13} \ P_{14} \ P_{15} \ \dots \ P_{1M}]$$

$$BiasCapaOculta = B1$$

$$BiasCapaSalida = B2$$

El **Genotipo**: consistirá en un arreglo de tamaño:

$$tamano = M * N + M + 2 \quad (1)$$

con la siguiente indexación:

- Las primetas  $M * N$  posiciones representarán la matriz de pesos de la capa oculta.
- Las siguientes  $M$  posiciones serán los pesos de la capa de salida.
- La penúltima posición será el bias de la capa oculta.
- La última posición será el bias de la capa de salida.

$$Cromosoma = [W_{11} \ \dots \ W_{1N} \ \dots \ W_{MN} \ P_{10} \ \dots \ P_{1M} \ B1 \ B2]$$

Ejemplo:

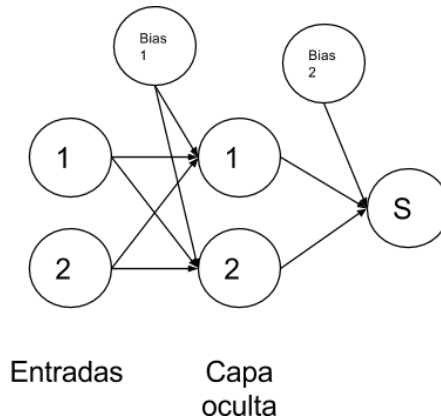


Figura 4: Arquitecturad de la red Neuronal de ejemplo.

Esta Red Neuronal de la figura 4 tiene dos entradas  $N = 2$ , dos Neuronas en la capa oculta  $M = 2$  y una salida. que servirá de ejemplo para explicar el algoritmo genético.

**El Fenotipo:**

$$PesosCapaOculto_{(2*2)} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

$$PesosCapaSalida_{(1*2)} = [P_{11} \quad P_{12}]$$

$$BiasCapaOculto = B1$$

$$BiasCapaSalida = B2$$

**El Genotipo:** El tamaño del cromosoma será por lo tanto

$$Tamano = M * N + M + 2 = 2 * 2 + 2 + 2 = 8$$

$$Cromosoma = [W_{11} \quad W_{12} \quad W_{21} \quad W_{22} \quad P_{11} \quad P_{12} \quad B1 \quad B2]$$

#### 4.2.2. Población:

La población consiste en determinar el número de cromosomas que habrán por generación, un mayor número de población determinará una mayor cantidad de variabilidad, pero también aumentará la complejidad del algoritmo, ya que esta es un factor muy influyente.

Tabla 2: Población

Individuo	Cromosoma
1	Cromosoma 1
2	Cromosoma 2
3	Cromosoma 3
4	Cromosoma 4
.	.
.	.
50	Cromosoma 50

#### 4.2.3. Población Inicial:

La población inicial en el algoritmo genético será totalmente aleatorio, aunque cumpliendo unas pequeñas reglas. En las pruebas hechas por la implementación de propagación hacia atrás se encontró que los pesos de la capa oculta varían en un rango de  $[-1, 1]$ , y los pesos de la capa de salida varían en un rango de  $[-7, 7]$ , dado esto la generación aleatoria cumplirá estos rangos en los pesos correspondientes. pero igualmente estos rangos no son camisa de fuerza para el algoritmos, ya que por medio de la reproducción y mutación se podrán salir de éstos si es necesario.

Siguiendo con el ejemplo, dos individuos de la población inicial, quedarían así:

$$Cromosoma1 = [-0,98 \quad 0,97 \quad -0,37 \quad 0,11 \quad -3,9 \quad 4,7 \quad 0,9 \quad 0,66]$$

$$Cromosoma2 = [-0,42 \quad 0,83 \quad -0,19 \quad 0,03 \quad -6,6 \quad -5,7 \quad 0,4 \quad 0,1]$$

#### 4.2.4. Selección por ruleta:

Consiste en asignarle a cada individuo un porcentaje de selección dependiendo de su fitness, siendo que para mejor fitness mayor será el porcentaje de selección y para menor fitness menor porcentaje de selección. organizando estos porcentajes en un arreglo, acumulando los porcentajes posición por posición, en donde la posición final el porcentaje debe ser 100 %, para que cuando la ruleta se ejecute (número random) recorrer el arreglo y encontrar el individuo seleccionado.

$$Ruleta = [0,01 \quad 0,03 \quad 0,19 \quad 0,26 \quad 0,30 \quad \dots \quad 0,94 \quad 1]$$

La posición de cada valor, representa a un individuo de la población, por lo tanto en donde caiga la ruleta, será el individuo seleccionado.

#### 4.2.5. Reproducción Directa:

consiste en darle un 50 % de probabilidad de intercambiar de gen al momento de la reproducción.

La reproducción de los cromosomas anteriores serían:

$$Cromosoma1 = [-0.98 \quad 0.97 \quad -0.37 \quad 0.11 \quad -3.9 \quad 4.7 \quad 0.9 \quad 0.66]$$

$$Cromosoma2 = [-0.42 \quad 0.83 \quad -0.19 \quad 0.03 \quad -6.6 \quad -5.7 \quad 0.4 \quad 0.1]$$

$$Hijo1 = [-0.98 \quad 0.83 \quad -0.37 \quad 0.11 \quad -6.6 \quad 4.7 \quad 0.9 \quad 0.1]$$

$$Hijo2 = [-0.42 \quad 0.97 \quad -0.19 \quad 0.03 \quad -3.9 \quad -5.7 \quad 0.4 \quad 0.66]$$

#### 4.2.6. Mutación Gaussiana:

Consiste en sumar un número que corresponde a un número aleatorio de una distribución normal con media en cero y desviación estandar 1.

Las probabilidades de mutación usadas en el proyecto son 8 % y 20 %, utilizando valores altos de mutación para tener una variabilidad alta.

$$Hijo2 = [-0.42 \quad 0.97 \quad -0.19 \quad 0.03 \quad -3.9 \quad -5.7 \quad 0.4 \quad 0.66]$$

$$Hijo2 = [-0.42 \quad 0.56 \quad -0.19 \quad 0.03 \quad -3.9 \quad -5.7 \quad 0.4 \quad 1.25]$$

#### 4.2.7. Reemplazo:

El reemplazo se realiza conservando un porcentaje de la población con mejor fitness, el porcentaje de la población que es reemplazada se puede determinar al inicio de la ejecución. Ejemplo:

- Para un población de 50 si se coloca un reemplazo de 88 %, la población reemplazada será 44 individuos.

- para una población de 50 si se coloca un reemplazo de 50 % la población reemplazada será 25 individuos.
- Para un población de 100 si se coloca un reemplazo de 94 %, la población reemplazada será 94 individuos.
- para una población de 100 si se coloca un reemplazo de 50 % la población reemplazada será 50 individuos.

#### 4.2.8. Función de evaluación:

Para la evaluación del cromosoma se va a utilizar el error cuadrático medio (ECM), donde un ECM alto significa que el cromosoma está dando respuestas muy alejadas a las esperadas y a menor ECM significa que el cromosoma está dando respuestas más certeras. La evaluación del cromosoma consiste en generar el fenotipo del cromosoma y éste cargarlo en la red neuronal, para que todos los datos del dataset sean evaluados y generar el error cuadrático medio.

$$ECM = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (2)$$

#### 4.2.9. Función de Aptitud o Fitness:

La función de aptitud debe ser una función que caracteriza a un cromosoma, entre mejor puntuación mejor será el cromosoma y a menor puntuación peor será el cromosoma, lo contrario del ECM; por lo tanto la función de aptitud será:

$$Aptitud = \frac{1}{ECM} \quad (3)$$

#### 4.2.10. Escalado:

Con intención de aumentar la variabilidad, y no guardar los mismos individuos generación tras generación. se considera implementar un escalado a todo el cromosoma a los individuos que no son reemplazados, para que en la próxima generación tengan un valor diferente pero no muy lejano.

Por ejemplo: si se usa un escalado del 1.1 tendríamos.

$$Cromosoma1 = [-0,98 \quad 0,97 \quad -0,37 \quad 0,11 \quad -3,9 \quad 4,7 \quad 0,9 \quad 0,66]$$

Después del escalado:

$$Cromosoma1 = [-1,078 \quad 1,067 \quad -0,407 \quad 0,121 \quad -4,29 \quad 5,17 \quad 0,99 \quad 0,726]$$

Si se usa un escalado de 1, quiere decir que se deja igual generación tras generación.

#### 4.2.11. Diagrama de flujo

El Diagrama de flujo del algoritmo sería el siguiente:

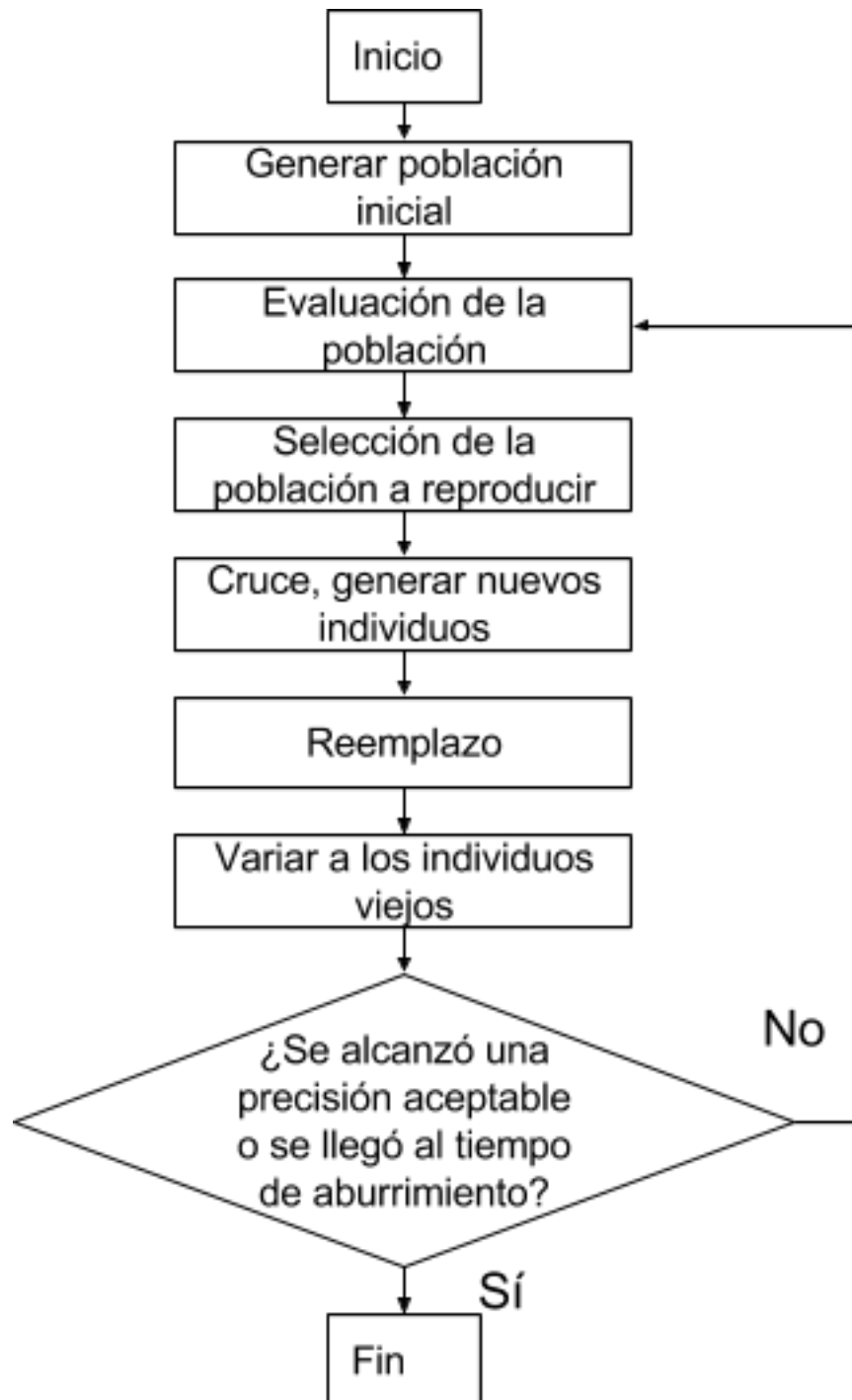


Figura 5: Diagrama de flujo del algoritmo.

## 5. Pruebas:

Al ejecutar las pruebas, los modelos y sus características son guardadas en archivos de texto. todos estos archivos de las pruebas se pueden encontrar en el repositorio del proyecto. URL: <https://github.com/hector2603/trabajoDeGrado/tree/practica> Además en la descripción del proyecto podrá encontrar la ubicación de las pruebas.

### 5.1. Backpropagation:

Para las Pruebas de backpropagation. El dataset usado para el entrenamiento de la red neuronal el 80 % de los datos serán para el entrenamiento y el 20 % para las pruebas, el dataset tiene 19 parámetro y después de su procesamiento tiene 55 por lo tanto éste será el número de datos de entrada.

Para el número de neuronas de la capa oculta se hizo pruebas con 19 y 38 neuronas.

Tabla 3: Precisión Pruebas Backpropagation

Neuronas	Prueba 1	Prueba 2	Prueba 3	Promedio Precisión
19	82.18 %	83.98 %	91.6 %	85.92 %
38	82.01 %	80.6 %	87.1 %	83.23 %

### 5.2. Algoritmos Genéticos:

Para las pruebas de los algoritmos genéticos se deben tener en cuenta diferentes variables que pueden generar una cantidad muy grande de combinaciones, por lo tanto éstas se deben limitar para generar un conjunto de combinaciones más discreto.

Tabla 4: Variables del algoritmo genético

Variables	Dato 1	Dato 2
Población	50	100
Reemplazo	Inserción	Élite
Mutación	8 %	20 %
Escalado	Sí	No

**Reemplazo Élite:** Corresponde en reemplazar un gran porcentaje de la población. 88 % en el caso de una población de 50 y 94 % en caso de una población de 100.

**Reemplazo Inserción:** Corresponde en reemplazar la mitad de la población, en cualquier caso del tamaño de la población, se reemplazará la mitad.

**Escalado:** Para estas pruebas se usó un escalado de 1,1.

Dada estas 4 variables, darían 16 combinaciones del algoritmo genético. que se representan en la siguiente tabla.

nombre	Poblacion	Mutacion	Reemplazo	Escalado	Prueba 1	Prueba 2
Prueba 1	50	8%	inserción	sí	0.7245873153779322	0.7245873153779323
Prueba 2	50	8%	inserción	no	0.7480451781059948	0.792354474370113
Prueba 3	50	8%	Élite	sí	0.7324066029539531	0.684622067767159
Prueba 4	50	8%	Élite	no	0.6898349261511729	0.738488271068636
Prueba 5	50	20%	inserción	sí	0.7237185056472633	0.684622067767159
Prueba 6	50	20%	inserción	no	0.6681146828844483	0.739357080799305
Prueba 7	50	20%	Élite	sí	0.7237185056472633	0.7341442224152911
Prueba 8	50	20%	Élite	no	0.7158992180712423	0.6811468288444831
Prueba 9	100	8%	inserción	sí	0.7454387489139879	0.7593397046046916
Prueba 10	100	8%	inserción	no	0.6663770634231103	0.8105994787141616
Prueba 11	100	8%	Élite	sí	0.6959165942658557	0.6950477845351868
Prueba 12	100	8%	Élite	no	0.6542137271937446	0.7132927888792354
Prueba 13	100	20%	inserción	sí	0.7237185056472633	0.6985230234578628
Prueba 14	100	20%	inserción	no	0.682015638575152	0.7758470894874022
Prueba 15	100	20%	Élite	sí	0.7176368375325803	0.6689834926151172
Prueba 16	100	20%	Élite	no	0.6785403996524761	0.6811468288444831

Figura 6: Pruebas algoritmos genéticos.



## 6. Conclusiones:

Tabla 5: Conclusiones.

Característica	Backpropagation	Algoritmos genéticos
Dependencia	Backpropagation depende mucho del primer modelo generado aleatorio, ya que utiliza derivadas para ir llegando a óptimos, puede quedar en óptimos locales.	Los modelos encontrados con algoritmos genéticos no dependen de las primeras generaciones, aumentando la variabilidad y no caer en óptimos locales.
Tiempo	Backpropagation puede tener una convergencia más rápida a comparación de los algoritmos genéticos, el problema es que si en algún momento cae en un óptimo local, por más tiempo que dure la ejecución nunca llegará a una solución.	Los algoritmos genéticos tienen una convergencia muy lenta, por lo cual el algoritmo puede demorar mucho más tiempo en llegar a un modelo que satisfaga el error permitido.
Resultados	En las diferentes pruebas de backpropagation, el primer modelo se generó con números aleatorios de 0 a 1 y en todos los modelos obtenidos al final se pudo caracterizar que los pesos de la primera capa varían $[-1,1]$ y los de la capa de salida de $[-7,7]$ .	Como ya se explicó en la implementación del algoritmo genético, la primera generación cumple una regla para generarla, que se determinó a partir de backpropagation, pero en los resultados obtenidos en todas las pruebas se pudo ver que los modelos ya no cumplían esta regla, lo cual verifica que los algoritmos genéticos tienen una gran variabilidad.
Precisión	Backpropagation alcanzó la mejor precisión con un 91 %.	Los algoritmos genéticos obtuvieron la mejor precisión con 81 %.

Como se puede evidenciar en la tabla, backpropagation alcanzó una mejor precisión, pero ésta también depende mucho de los datos iniciales, no es explorativa y cae en óptimos locales. Por otra parte los algoritmos genéticos obtuvieron una menor precisión, pero dado a las cualidades de éstos, se puede explorar más los espacios de búsqueda, no caer en óptimos locales. Así mismo se puede decir que backpropagation da respuestas con buena precisión y usables, pero el algoritmo evolutivo está más cerca de llegar al 100 % de precisión.

## 7. Referencias

- [1] A. J. Rosales-silva and F. J. Gallegos-funes, “Detection and classification of Non-Proliferative Diabetic Retinopathy using a Back-Propagation neural network Detección y clasificación de Retinopatía Diabética no Proliferativa usando una Red Neuronal de Retropropagación,” pp. 70–85, 2015.
- [2] B. Antal and A. Hajdu, “An ensemble-based system for automatic screening of diabetic retinopathy,” *Knowledge-Based Systems*, vol. 60, pp. 20–27, 2014.
- [3] T. Taveira-Gomes, “machine learning on the diabetic retinopathy debrecen data set data set.” <https://rpubs.com/tiagotaveira/debrecen>.
- [4] S. J. J. Kumar and M. Madheswaran, “An improved medical decision support system to identify the diabetic retinopathy using fundus images,” *Journal of Medical Systems*, vol. 36, no. 6, pp. 3573–3581, 2012.
- [5] S. Issac Niwas, P. Palanisamy, R. Chibbar, and W. J. Zhang, “An expert support system for breast cancer diagnosis using color wavelet features,” *Journal of Medical Systems*, vol. 36, no. 5, pp. 3091–3102, 2012.
- [6] E. Elveren and N. Yumuşak, “Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm,” *Journal of Medical Systems*, vol. 35, no. 3, pp. 329–332, 2011.
- [7] V. Cruz-Gutiérrez and A. Sánchez-López, “Un sistema experto difuso en la Web para diagnóstico de diabetes,” vol. 107, pp. 145–155, 2015.
- [8] L.Álvarez Menéndez@, F. J. de Cos Juez, F.ánchez Lasheras@, and J. A. Álvarez Riesgo, “Artificial neural networks applied to cancer detection in a breast screening programme,” *Mathematical and Computer Modelling*, vol. 52, no. 7-8, pp. 983–991, 2010.
- [9] R. Kunhimangalam, S. Owallath, and P. K. Joseph, “A clinical decision support system with an integrated EMR for diagnosis of peripheral neuropathy,” *Journal of Medical Systems*, vol. 38, no. 4, 2014.
- [10] M. S. Bascil and H. Oztekin, “A study on hepatitis disease diagnosis using probabilistic neural network,” *Journal of Medical Systems*, vol. 36, no. 3, pp. 1603–1606, 2012.
- [11] M. S. Bascil and F. Temurtas, “A study on hepatitis disease diagnosis using multilayer neural network with Levenberg Marquardt training algorithm,” *Journal of Medical Systems*, vol. 35, no. 3, pp. 433–436, 2011.
- [12] B. D. Barkana, I. Saricicek, and B. Yildirim, “Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ANN, SVM, and classifier fusion,” *Knowledge-Based Systems*, vol. 118, pp. 165–176, 2017.

- [13] M. Javidi, H.-R. Pourreza, and A. Harati, “Vessel segmentation and microaneurysm detection using discriminative dictionary learning and sparse representation,” *Computer Methods and Programs in Biomedicine*, vol. 139, pp. 93–108, 2017.
- [14] F. Izaurieta and C. Saavedra, “Redes neuronales artificiales,” *Departamento de Física, Universidad de Concepción Chile*, 2000.
- [15] T. Zhang, “An introduction to support vector machines and other kernel-based learning methods,” *AI Magazine*, vol. 22, no. 2, p. 103, 2001.
- [16] A. E. Eiben, J. E. Smith, *et al.*, *Introduction to evolutionary computing*, vol. 53. Springer, 2003.