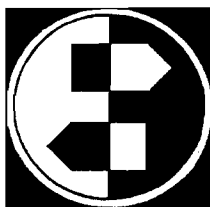
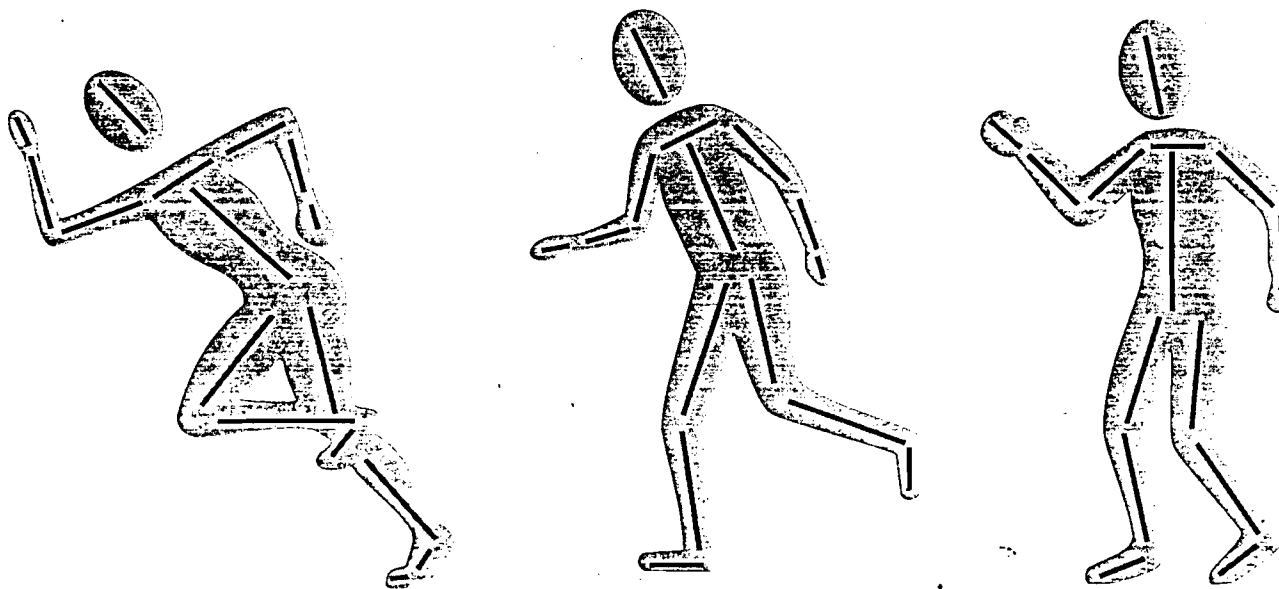


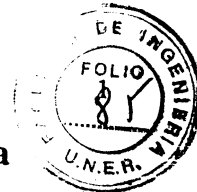
# Revista Argentina de Bioingeniería



Volumen 2, Nro 2, Setiembre 1996



Sociedad Argentina de Bioingeniería



## Autoridades S.A.B.I.

### Presidente

Ing. Jorge E. Monzón

### Vicepresidente

Dr. Ing. Ricardo L. Armentano

### Secretaria

Prof. Teresa R. de Montaner

### Tesorero

Ing. Ricardo A.M. Taborda

### Vocal 1

Dra. Estela Ruiz

### Vocal 2

Ing. Héctor G. Polimeni

### Presidente anterior

Dr. Peter Willshaw

## Revista Arg.de Bioingeniería

### Editor

Ricardo A. M. Taborda

### Cuerpo editorial

??

La Revista Argentina de Bioingeniería es una publicación semestral de la Sociedad Argentina de Bioingeniería (S.A.B.I.), no está disponible para su venta al público y solo puede obtenerse mediante membresía a la Sociedad. Para instrucciones de autor y/o para mayor información, rogamos dirigirse a la regional de S.A.B.I. mas cercana cuyas direcciones y demas datos están listados en la contratapa.

## Revista Argentina de Bioingeniería

Septiembre 1996

Volumen 2

Número 2

Pag.	Contenido
2	<b>Mensaje del Presidente</b> <i>Difusión</i>
3	<b>Lista de correo electrónico sobre Bioingeniería</b> Gonzalo Montenegro
5	<b>Biomateriales: Polímeros biomédicos</b> Teresita R. Cuadrado <i>Trabajos originales</i>
11	<b>Diagnóstico de cardiopatías mediante redes neuronales y algoritmos genéticos</b> J. Goddard, I. López y H.L. Rufiner
20	<b>Técnica de videografía digital aplicada al análisis de la marcha humana</b> M. Cleva, J. Aguilar, J.E. Monzón
29	<b>Potencia muscular en movimientos de flexoextensión del sistema tronco-muslo humano</b> C. Yamin Turbay, G. Goroso, O. Bravo
39	<b>Variabilidad de las señales cardiorespiratorias.</b> <b>Parte 2: Variabilidad a largo plazo</b> M. Risk, J. Sobh, R.Barbieri, A.Ramírez y J. Philip Saúl <i>Rincón educativo</i>
46	<b>La oferta educativa de grado en la Argentina: La carrera de Bioingeniería en la Facultad de Ingeniería, U.N. SanJuan</b> Héctor Gustavo Polimeni <i>Cursos / Congresos / Conferencias</i>
48	<b>Congreso SABI'96</b> <i>Anuncios / Novedades</i> Están por todas partes, descúbralos ! Entre otros:
19	<b>Premio Max'96</b>
28	<b>Número conjunto SABI-FAC - Mas instrucciones de autor...</b>
45	<b>CORAL VII, Niza'97 y otros eventos</b>

Los artículos firmados se publican bajo exclusiva responsabilidad de sus autores. El material publicado se puede reproducir o citar previa autorización por escrito del editor. En todos los casos se deberá indicar su procedencia, los autores, y remitir dos ejemplares de la transcripción al editor.

## Diagnóstico de Cardiopatías mediante Redes Neuronales y Algoritmos Genéticos

John Goddard<sup>(\*)</sup>, Irineo López<sup>(\*\*)</sup>, Hugo Leonardo Rufiner<sup>(\*)(#)</sup>

(\*)UAMI, Departamento de Ing. Eléctrica, Av. Michoacán y La Purísima S/N,  
(09340), México D.F., México, jgc@xanum.uam.mx.

(\*\*)UACH, Centro de Computo Avanzado, Chapingo, México, ilopez@mail.internet.com.mx.

(#)UNER, Laboratorio de Cibernética, Ruta 11 Km 10, Paraná,  
(3100), Entre Ríos, Argentina, rufi@xanum.uam.mx

### Resumen

Durante los últimos años han aparecido un número creciente de artículos dedicados a la aplicación de técnicas de inteligencia artificial a la medicina. En particular, el problema de clasificación automática ha recibido una atención especial. En el presente trabajo se utilizan redes neuronales para el diagnóstico automático de patologías cardíacas. Se realiza una comparación entre distintas formas de hallar sus parámetros, a saber: retropropagación, programas evolutivos, y alternativas híbridas. El uso de programas evolutivos tiene la ventaja de que permite aprender por medio de una regla uniforme y con una libertad de elección de parámetros que no está permitida en el algoritmo de retropropagación.

**Palabras Claves:** • Programas Evolutivos • Redes Neuronales • Clasificación • Corazón

### Introducción

La inteligencia artificial es una disciplina joven, de cuarenta años de edad, que según una pseudo definición<sup>1</sup> es "como hacer que las computadoras resuelvan problemas que los seres humanos realizan mejor en este momento". Un problema de este tipo que surge en la medicina es el relacionado con el diagnóstico médico. El problema se centra en la habilidad de que un programa computacional pueda reconocer si un paciente padece de una enfermedad, dada la información pertinente sobre sus signos vitales. Esto implica que el programa debe contener el conocimiento médico necesario para llegar a una conclusión adecuada.

Se han desarrollado varias técnicas para obtener automáticamente ese conocimiento, como parte de una rama de la inteligencia artificial denominada *aprendizaje maquina*. Usualmente estas técnicas requieren de un conjunto de casos previamente clasificados señalando los valores de las variables pertinentes y la clase a la cual pertenece cada uno. En el caso médico, se trataría de un conjunto de casos clínicos con el diagnóstico correspondiente. El objetivo de la técnica es utilizar la información de estos casos para encontrar relaciones entre las variables involucradas y construir una regla de clasificación. Esta forma de aprendizaje se denomina *aprendizaje supervisado*<sup>2</sup>.

Entre las técnicas de aprendizaje supervisado se encuentran: la inducción automática de árboles de decisión o reglas como ID3, C4.5 y CART<sup>2,3,4,5</sup>, las redes neuronales<sup>6,7,8</sup>, y las técnicas híbridas<sup>9,10,11,12,13,14</sup>. Un importante estudio del proyecto ESPRIT de la Comunidad Europea, compara algunas de estas técnicas entre sí, y con métodos estadísticos, sobre distintos conjuntos de datos<sup>15</sup>. Sin embargo, el área de la medicina presenta dificultades que no se encuentran en dominios más precisos. Tal es el caso de los datos inconsistentes (diagnósticos opuestos para casos similares), o faltantes (faltan valores de algunos atributos o variables). Estos problemas se deben en gran parte a que los datos son generalmente tomados de historias clínicas.

ID3, C4.5 y CART utilizan los casos para construir, en una sola pasada, un árbol de decisión cuyos nodos internos corresponden a decisiones de las variables involucradas, y cuyas hojas representan una clasificación. Con frecuencia, estos árboles se podan para permitir una mejor generalización sobre casos que no vistos. En ese caso las hojas podrán tener una probabilidad de clasificación asociada.

Debido a su estructura, los árboles de decisión particionan el espacio de las variables en hiper-planos paralelos a los ejes. Esto puede ser demasiado restrictivo para cierto tipo de problema. A diferencia, los perceptrones multicapa (PMC), un tipo de red neuronal, tienen la capacidad de aproximar regiones de decisión arbitrariamente complejas, y además tolerar mejor los problemas antes mencionados (por ejemplo los datos ruidosos)<sup>6,16,17</sup>.

Se han propuesto varios algoritmos para entrenar las redes neuronales, sin embargo el más popular, retropropagación (BP), está basado en el método de gradiente descendente, que en mayor o menor grado puede sufrir del problema de caer en mínimos locales<sup>2</sup>. Recientemente se han utilizado técnicas basadas en algoritmos genéticos (GA) y programas evolutivos (PE) para el entrenamiento de redes neuronales con muy buenos

resultados<sup>18</sup>. Su naturaleza paralela permite hacer una extensa búsqueda y evitar los mínimos locales. Además permite gran flexibilidad en cuanto a la definición de la función a maximizar permitiendo modificarla según el problema<sup>19</sup>.

En el presente artículo se presentan los resultados obtenidos de aplicar los PMC a un problema de clasificación médica, mediante diferentes formas de entrenamiento. En la sección siguiente se presenta el paradigma de las redes neuronales y retropropagación, seguido por una sección acerca de algoritmos genéticos y programas evolutivos. A continuación se describe el conjunto de datos de desordenes cardíacos utilizado, y posteriormente los métodos empleados para entrenar las redes. En la sección siguiente se describen los resultados obtenidos, y finalmente se presenta una discusión y las conclusiones derivadas de esta investigación.

## Redes Neuronales

Las redes neuronales artificiales (RNAs) intentan simular, al menos parcialmente, la estructura y funciones del cerebro y del sistema nervioso de los seres vivos. Una RNA es un sistema de procesamiento de información o señales compuesto por un gran número de elementos simples de procesamiento, llamados neuronas artificiales o simplemente nodos. Dichos nodos están interconectados por uniones directas llamadas conexiones y cooperan para realizar procesamiento en paralelo con el objetivo de resolver una tarea computacional determinada. Muchos modelos de RNAs han sido desarrollados para una variedad de propósitos. Cada uno de ellos difieren en estructura, implementación y principio de operación, pero a su vez tienen características comunes<sup>2</sup>.

El PMC (*Figura 1*) consiste en un arreglo de nodos ubicados en capas, de forma tal que los nodos de una capa están conectados a todos los nodos de la capa anterior y de la siguiente mediante los pesos de conexión. La primera capa se denomina capa de entrada y la última se denomina capa de salida. Las capas que quedan entre estas dos se denominan capas ocultas. No existe un límite para fijar la cantidad de capas de un PMC, pero se ha demostrado que un PMC con una capa oculta y con un número suficiente de nodos es capaz de solucionar casi cualquier problema<sup>16</sup>.

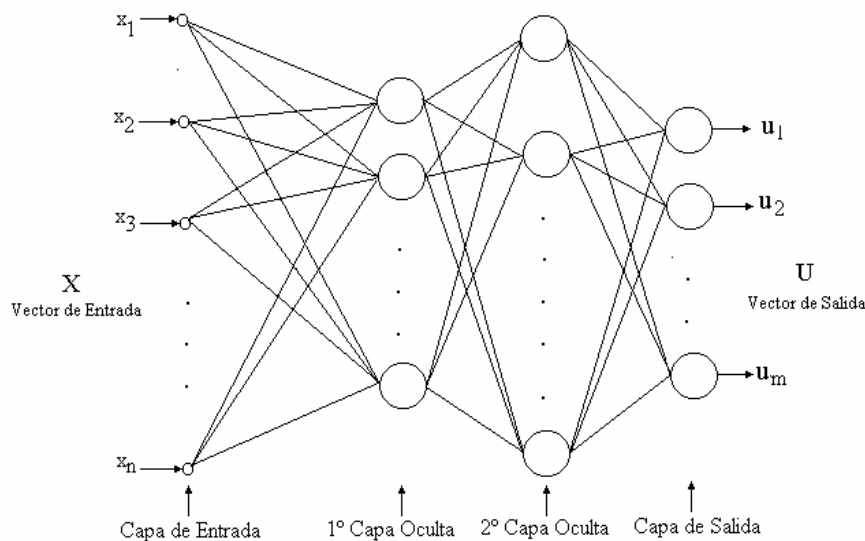


Figura 1: PMC con 2 capas ocultas

Una de las características atractivas de los PMCs es su capacidad para auto-adaptarse cambiando sus fuerzas de conexión para encontrar una combinación de pesos adecuados para clasificar los casos correctamente. Esto se logra usualmente por medio de BP.

Para calcular el valor de salida de un nodo en una capa oculta o de salida, se suman las entradas al nodo multiplicando cada una por el peso correspondiente a su conexión. Posteriormente se aplica una función de activación, usualmente del tipo sigmoidea. En la capa de entrada, los valores pasan directamente a la siguiente capa sin cambiarlos. Para entrenar el PMC generalmente se utiliza el algoritmo de BP<sup>2</sup>. Durante el entrenamiento, se presenta el valor de las variables de entrada para cada caso (similar a lo mencionado en la introducción), y la respuesta de cada nodo en la capa de salida es comparada con la correspondiente respuesta deseada. Los errores asociados son calculados para adaptar los pesos en cada capa por medio de un método de gradiente descendente. Como consecuencia de este método, la función de activación de los nodos debe ser diferenciable. Este proceso es repetido hasta que el error entre los valores de salida y los deseados, para cada

uno de los casos, llega a un nivel aceptable. En síntesis, los pasos implicados en el aprendizaje serían los siguientes:

```

procedimiento retropropagación
{
  inicializar pesos y umbrales
  mientras (criterio de terminación no se alcance)
  {
    seleccionar ejemplo
    aplicar a la entrada de la red
    calcular salida de la red
    calcular error entre la salida y su valor deseado
    ajustar pesos y umbrales para minimizar el error
  }
}

```

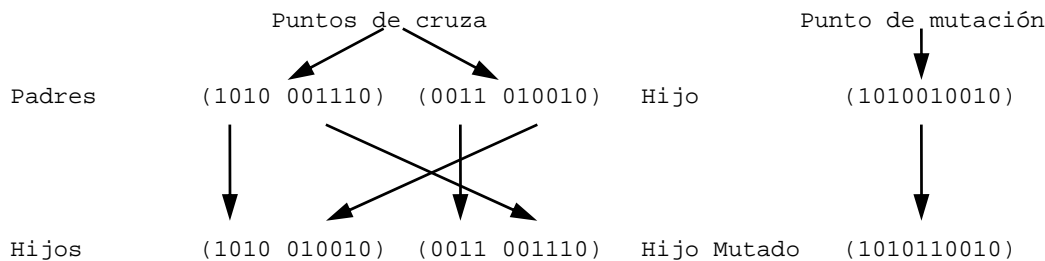
La idea fundamental detrás del método de gradiente descendente puede verse a través de una analogía con una bola que se desliza por una superficie. La altura de la superficie representa el error para cada punto en el espacio de los pesos. De esta manera la bola se desliza por la superficie siguiendo el sentido del gradiente descendente de la misma. Como se puede apreciar de la analogía, la posición inicial de la bola es de suma importancia. Si estamos en una zona relativamente plana de la superficie tardaremos mucho en hallar un mínimo. Si estamos cerca de un mínimo local, es muy probable que caigamos en él. Para solucionar parcialmente este problema se agrega al algoritmo un término de inercia o momento. También se puede repetir varias veces el procedimiento iniciándolo en distintos puntos (generalmente cerca del origen) y guardar los pesos de la corrida que llegue a menor error final. Para más detalles se puede consultar la extensa bibliografía al respecto (por ej. 6,7,8).

Para utilizar un PMC entonces, hay que definir su estructura en términos de nodos y capas, que adecuadamente representa el problema bajo consideración. Como se vió el algoritmo de BP puede demorarse para llegar a un mínimo aceptable, y por ser un método de gradiente descendente, es posible que converja a un mínimo local.

### Algoritmos Genéticos y Programas Evolutivos

Los GAs son métodos adaptativos que pueden ser utilizados para resolver problemas complejos de optimización. Como en el caso de las redes neuronales, están inspirados en la naturaleza, y aquí en las teorías de los procesos genéticos de los organismos biológicos.

La analogía en que se basan los GAs, estriba en reconocer que el mecanismo esencial del proceso evolutivo en la naturaleza, radica en los cromosomas, el material genético o el conjunto de genes de un individuo biológico, lo cual determina sus características únicas. Los cambios en el material genético de las especies permiten el proceso de adaptación. Las fuerzas que subyacen el proceso evolutivo son la conjunción de la selección natural, la recombinación de material genético y la mutación, fenómenos que se presentan durante la reproducción de las especies. Así como competencia entre los individuos por los recursos naturales limitados y por la posibilidad de procreación o reproducción, lo que permite que solo los más fuertes o más adaptados sobrevivan. Esto significa que el material genético de los mejores individuos sobrevive y se reproduce, mientras que los genes de los individuos más débiles o menos adaptados, mueren o se extinguen.



En 1975, Holland<sup>20</sup> propuso los GAs como una técnica computacional que intenta imitar el proceso evolutivo de la naturaleza para el diseño de sistemas artificiales adaptativos. En esta realización computacional, los GAs manipulan una población de soluciones potenciales codificadas en cadenas binarias. El conjunto de cadenas representan el material genético de una población de individuos. Los operadores correspondientes de

selección, cruzamiento, y mutación son aplicados iterativamente para buscar los mejores individuos o las mejores soluciones. Cada solución potencial se asocia con un valor de aptitud, que mide que tan buena es comparada con las otras soluciones de la población.

Un algoritmo que efectúa este proceso es<sup>21</sup>:

```

procedimiento algoritmo genético
{
    inicializar población
    evaluar población
    mientras (criterio de terminación no se alcance)
    {
        seleccionar soluciones para la siguiente población
        ejecutar cruzamiento y mutación
        evaluar población nueva
    }
}

```

Usualmente, se inicializa la población aleatoriamente, y mediante la función de aptitud se evalúa la población. Con frecuencia se aplica el método denominado *roulette*, que, para escoger una nueva población del mismo tamaño, da un peso mayor a las soluciones con mayor aptitud. Luego se efectúa el cruzamiento o intercambio de genes, el cual básicamente consiste en cambiar segmentos de los cromosomas entre parejas seleccionadas en forma aleatoria (Figura 2). Posteriormente, un gen puede mutarse, es decir un *bit* puede cambiarse, en forma aleatoria, de acuerdo al valor del parámetro *probabilidad de mutación* del algoritmo genético (Figura 3). Este proceso continua iterativamente, hasta que se alcance algún criterio, obteniendo una solución. Existen algunos resultados teóricos<sup>22</sup> acerca de si la solución obtenida es la óptima, la importancia de los operadores usados, y la mejor elección de los parámetros involucrados. Sin embargo todavía hay bastante discusión y no hay repuestas definitivas en todos de los casos. Más aún, recientemente se está generalizando el empleo de estructuras más complejas de representación<sup>23</sup>, como el uso de números reales en lugar de números binarios. Los operadores genéticos también sufren adecuaciones cuando se modifica la representación de las soluciones. A estos algoritmos se los denomina de Programas Evolutivos (PEs). En adelante utilizaremos indistintamente PE o GA para referirnos a los algoritmos. En la sección de métodos se explica la manera de aplicar los PEs al entrenamiento de un PMC. Una ventaja de esta forma de entrenamiento es que se están evaluando en forma simultánea varios puntos del espacio de pesos con la consiguiente evasión de los mínimos locales.

### Descripción de los Datos

En este trabajo se utilizó un conjunto de datos de desordenes cardíacos compilada originalmente en la Cleveland Clinic Foundation y suministrada por Robert Detrano, M.D., Ph. D. del V.A. Medical Center, Long Beach, CA. Es parte de una colección de bases de datos de la Universidad de California, recolectadas por David Aha<sup>24</sup>.

El propósito de este conjunto de datos es predecir la presencia o ausencia de enfermedad cardíaca dados los resultados de varias pruebas médicas llevadas a cabo sobre el paciente. La base de datos posee 13 atributos (ver la Tabla 1), que fueron extraídos de un conjunto mayor de 75. El conjunto de datos original contenía 303 ejemplos pero algunos fueron descartados quedando finalmente 270. Existen dos clases: presencia o ausencia de enfermedad cardíaca. Esto constituye una reducción del número de clases en los datos originales, en los cuales había cuatro grados diferentes de enfermedad cardíaca. En la Tabla 2 se puede observar la distribución de los ejemplos para cada clase. El hecho de que exista una mayor cantidad de casos de pacientes sanos y que estos pacientes sanos sean más 'fáciles' de clasificar plantea problemas especiales. Esto se puede resolver utilizando un costo diferente para los errores en cada clase<sup>15</sup> o incorporando estos costos en BP o GAs<sup>19</sup>.

Esta base de datos ha sido utilizada previamente en la literatura. Específicamente en el proyecto "StatLog" bajo el programa ESPIRIT de la Comunidad Europea<sup>15</sup>, mencionado en la introducción. Sus resultados mostraron que los métodos estadísticos se comportaban mejor para este problema. Entre los métodos basados en árboles CART obtuvo el mejor desempeño. BP está en orden 16 dentro de sus resultados. En cuanto a tiempos, BP figura a la mitad entre los métodos probados. En el trabajo no aparecen resultados con respecto a GA.

**Tabla 1** : Información acerca de los Atributos utilizados

Nº	Descripción	Tipo
1	Edad	Real
2	Sexo	Binario
3	Dolor de Pecho (4 valores)	Nominal

4	Presión Sanguínea en Reposo	Real
5	Colesterol (mg/dl)	Real
6	Nivel de azúcar en sangre > 120 mg/dl (en ayunas)	Binario
7	Resultados electrocardiográficos en Reposo (valores 0,1,2)	Nominal
8	Máxima frecuencia cardíaca alcanzada	Real
9	Angina Inducida por ejercicio	Binario
10	Depresión del segmento ST inducida por ejercicio con respecto al reposo	Real
11	Pendiente del segmento ST (pico del ejercicio)	Ordenado
12	Nº de vasos mayores coloreados por fluoroscopia (0-3)	Real
13	Thal.: 3 = normal; 6 = defecto fijo; 7 = defecto reversible	Nominal

**Tabla 2:** Distribución de las clases en los datos

Clase	Ejemplos
Sanos	150 (55.56%)
Enfermos	120 (44.44%)
<b>Total</b>	<b>270 (100.0%)</b>

## Métodos

A continuación describiremos los detalles de implementación y las condiciones utilizadas en los experimentos realizados sobre el conjunto de datos elegido. Se realizaron tres tipos de experimentos. El primero fue realizado entrenando la red con BP. Para el segundo caso se utilizaron PE con tres diferentes funciones de aptitud y finalmente se probaron cuatro formas diferentes de hibridar BP y PE. La red neuronal utilizada poseía 13 entradas, 13 neuronas en la capa oculta y una salida. En todos los casos los algoritmos utilizados ajustaron pesos y umbrales.

Para BP se utilizó el algoritmo mejorado que incluye el momento de los pesos (BP 990 en los resultados). Esta característica ha demostrado evitar algunos mínimos locales. En los casos en que se utilizó la alternativa híbrida, el momento no fue necesario debido a que la parte genética se encarga de evitar los mínimos locales.

La codificación de la red utilizada para el PE fue directa, es decir, cada cromosoma estuvo constituido por una cadena de números reales que correspondían a los pesos y umbrales de la red. Se utilizó el método de selección de individuos denominado *tournament* que consiste en comparar la aptitud de N individuos y elegir el mejor como progenitor de la siguiente generación. En este caso se tomaron tres individuos. El operador de cruzamiento utilizado fue el de *cruza doble* con una probabilidad de 1. El operador de mutación empleado fue el de *mutación con bias gaussiano*, que consiste en sumar un valor aleatorio de distribución gaussiana a cada gen. Este tipo de mutación tiene la ventaja de producir variaciones pequeñas con mayor probabilidad resultando en pocos saltos abruptos (que también son necesarios pero pueden afectar la convergencia si aparecen muy seguido). La amplitud de esta mutación se ajustó para decaer exponencialmente con las generaciones. Todos los genes se mutaron (probabilidad 1). También se utilizó una estrategia denominada *Elitismo* que permite rescatar al mejor individuo de cada generación e incluirlo en la siguiente. Este enfoque generalmente mejora los resultados porque evita la pérdida de la mejor solución.

Las funciones de aptitud utilizadas para el PE fueron las siguientes:

$$ECM = \frac{\sqrt{\sum error^2}}{\# patrones}$$

$$EAM = \frac{\sum |error|}{\# patrones}$$

$$MAL = \# mal clasificados$$

Estas corresponden en los resultados a las columnas etiquetadas como GA ECM, GA EAM y GA MAL.

Para el caso híbrido se utilizaron parámetros diferentes para la parte evolutiva debido a que el objetivo era que el PE hiciera una búsqueda más extensa en la superficie a minimizar y que posteriormente BP bajara hasta el mínimo. Por este motivo se utilizó *mutación con bias uniforme* y *cruza simple* o *uniforme* según el caso. Debido a este cambio las probabilidades de cruzar y mutación se mantuvieron por debajo de 1 (no se aplicaron sobre todos los individuos y genes). Estas combinaciones parecieron funcionar mejor en pruebas preliminares. Se probaron solo dos de las funciones de aptitud anteriores: ECM y MAL. Los dos primeros experimentos (HIB

ECM 2 y HIB MAL) consistieron en obtener generaciones mediante cruza y selección y luego aplicar BP durante una cantidad de épocas determinada a cada individuo para finalmente evaluar la aptitud de cada uno mediante las funciones antedichas. Este es un enfoque *Lamarckiano* de la evolución ya que los caracteres adquiridos se transmiten a la siguiente generación. Sin embargo lo que nos interesa aquí no es su correlato biológico sino la efectividad del enfoque para resolver nuestro problema. La cantidad de épocas x individuo x generación se mantuvo en 0.3 (30% de los ejemplos de entrenamiento) por razones de equiparación como se explica más adelante. El tercer enfoque (HIB ECM 1) consistió en variar la cantidad de épocas x individuo durante el entrenamiento de manera de que al principio solo actuará el algoritmo genético y al final actuará más BP. Esto se logra también con la disminución progresiva de la amplitud de las mutaciones. La última alternativa (GA+BP) consistió en utilizar primero el PE solo para encontrar un *buen punto de partida* para BP. El PE se corrió durante 150 generaciones con 10 individuos, cruza uniforme y amplitud de mutación decreciente. el resto de los parámetros se mantuvieron igual que en las corridas anteriores. Con el mejor individuo se inicializó BP para correr durante 495 épocas (otra vez por razones de equiparación), con un parámetro de ajuste de pesos de 0.2 (para evitar olvidar lo aprendido por el PE). Para más detalles sobre los operadores y su utilización <sup>23,25</sup>.

Para realizar mejor la comparación de los distintos algoritmos se utilizó un método de estimación de los errores de entrenamiento y prueba denominado Validación Cruzada (VC). Este método consiste en dividir el conjunto inicial de datos en N particiones aleatorias. De esta manera se realizan las pruebas sobre cada partición y luego se promedian los resultados. Esto evita el sesgo que se produce en los resultados al hacer una sola partición aleatoria como se acostumbra frecuentemente. El problema de este enfoque es que los tiempos de evaluación se multiplican por N y esto puede ser excesivo en algunos casos hasta para Ns relativamente pequeños. En nuestro caso elegimos N=5 pesando las consideraciones anteriores, y la distribución de los ejemplos por clase en cada partición puede observarse en la Tabla 3. Aquí los archivos \*.TRN corresponden a los utilizados para entrenamiento y los \*.TST son los de prueba o validación. Todos los datos fueron normalizados con respecto al conjunto completo.

**Tabla 3:** Particiones de los datos para validación cruzada

Archivo	Sanos (%)	Enfermos (%)	Cantidad Total
H1.TRN	53.0	47.0	217
H1.TST	66.0	34.0	53
H2.TRN	55.5	44.5	218
H2.TST	55.8	44.2	52
H3.TRN	56.2	43.8	210
H3.TST	53.3	46.7	60
H4.TRN	54.3	45.7	219
H4.TST	60.8	39.2	51
H5.TRN	58.8	41.2	216
H5.TST	42.6	57.4	54

Todos los resultados fueron obtenidos corriendo los algoritmos por 300 generaciones de 10 individuos o el equivalente para los otros métodos. Para esto se calculó a cuantas épocas correspondía una generación de un solo individuo, de acuerdo al tiempo y la cantidad de operaciones utilizado por los algoritmos en igualdad de condiciones. Esta cifra fue aproximadamente :

$$\frac{1}{3} \cdot \frac{\text{épocas}}{\text{generación} \times \text{individuos}}$$

De esta manera BP se corrió durante 990 épocas y para el algoritmo híbrido de épocas fijas se utilizaron 150 generaciones de 10 individuos con 0.3 épocas x individuo x generación. En el caso de épocas variables se utilizaron 100 generaciones de 10 individuos, partiendo de 0 épocas (genético puro) hasta llegar a una época x generación x individuo al final de la corrida.

## Resultados

A continuación se presentan los resultados del entrenamiento de los distintos algoritmos. En la Tabla 4 se muestran los resultados de BP y GA corridos en forma separada. Como se observó sobreaprendizaje de la red en 990 épocas, se repitió el experimento para una cantidad de épocas menor (250). En la Tabla 5 se muestran los resultados de las distintas alternativas híbridas según se explicaron en la sección anterior. En todos los casos se presentan los porcentajes de aciertos de cada método para el conjunto de prueba (TST) y el de entrenamiento



(TRN) y se muestran valores promedio sobre las particiones del método de validación cruzada. En la *Figura 4* se presenta un gráfico comparativo que facilita el análisis de estos resultados.

**Tabla 4:** Resultados BP y GA

	BP 250		BP 990		GA MAL		GA EAM		GA ECM	
	TRN	TST	TRN	TST	TRN	TST	TRN	TST	TRN	TST
<b>Sano</b>	88.9	76.0	90.7	70.5	81.1	79.8	89.2	79.9	88.2	75.7
<b>Enfermo</b>	84.3	73.4	90.7	68.0	85.2	73.8	83.9	71.8	75.2	65.3
<b>Total</b>	86.8	73.9	90.8	69.0	88.4	77.5	86.9	75.8	80.6	71.1

**Tabla 5:** Resultados GA+BP

	HIB MAL		HIB ECM 2		HIB ECM 1		GA+BP	
	TRN	TST	TRN	TST	TRN	TST	TRN	TST
<b>Sano</b>	90.5	78.2	86.8	82.4	90.4	83.1	96.7	78.0
<b>Enfermo</b>	86.6	74.7	84.7	80.4	83.0	78.4	87.8	70.6
<b>Total</b>	88.8	76.1	86.0	81.0	87.0	80.1	92.7	74.4

## Discusión y Conclusiones

En este trabajo hemos investigado la aplicación de PE al entrenamiento de un PMC para un problema de diagnóstico médico.

Como se había visto en<sup>15</sup> los PMC entrenados por BP no demostraron buen desempeño en relación a otras técnicas. Esto coincide con los resultados presentados en la sección anterior. Aunque los PMC se comportaron muy bien con los datos de entrenamiento, mostraron muy mala capacidad de generalización sobre los datos de prueba.

Se puede observar que se obtiene la mejor generalización con las técnicas híbridas, que combinan la exploración simultánea de varios puntos del espacio debida a los PE, con la velocidad del descenso del método de gradiente descendente cuando se encuentra en las cercanías del mínimo. Una posible desventaja de estas técnicas sería el costo computacional implicado. Sin embargo en este caso se equipararon los tiempos de todos los algoritmos (salvo BP 250), aunque los recursos de memoria empleados por los PE serían mayores (proporcional con el número de individuos).

Otro problema asociado al algoritmo de BP es que los porcentajes de reconocimiento sobre los datos de entrenamiento varían de época en época. Esto hace difícil determinar un criterio de terminación. La alternativa más usada es terminar cuando el ECM se hace menor que un valor prefijado. Sin embargo esto no implica necesariamente un mayor porcentaje de respuestas correctas sobre el conjunto de entrenamiento. Más aún, la capacidad de generalización empeora cuando disminuye demasiado el ECM (sobrepone los ejemplos), según se desprende de los resultados de BP 250 y 990. La posibilidad de minimizar otras funciones como la cantidad de respuestas erróneas mediante EP ofrece otras alternativas (ver resultados de GA MAL).

Otra ventaja de los PE no utilizada en este artículo es la de poder evolucionar la estructura de la red junto con los pesos<sup>18</sup>, y también su capacidad para seleccionar los atributos más importantes para el diagnóstico<sup>26</sup>. Esto potenciaría aún más las técnicas planteadas pero requeriría utilizar una codificación con distintos tipos de datos o un *meta algoritmo genético* (donde cada cromosoma es un GA).

El empleo de validación cruzada permite mejorar la confiabilidad de la tasa de error obtenida. Un método comúnmente utilizado es tomar una sola partición para obtener los errores de entrenamiento y prueba. Sin embargo aún en el caso de tomar solo cinco particiones hemos observado una variación muy grande en los resultados. Por ejemplo hemos hallado hasta un 40% de diferencia en el error de prueba para un mismo método sobre distintas particiones. De no utilizar algún método de estimación del error como este se corre el riesgo de que alguna partición favorezca alguno de los métodos invalidando todos los resultados.

Agradecimientos: Este trabajo fue parcialmente realizado utilizando el paquete Sugal, escrito por el Dr. Andrew Hunter de la Universidad de Sunderland, Inglaterra<sup>25</sup>.

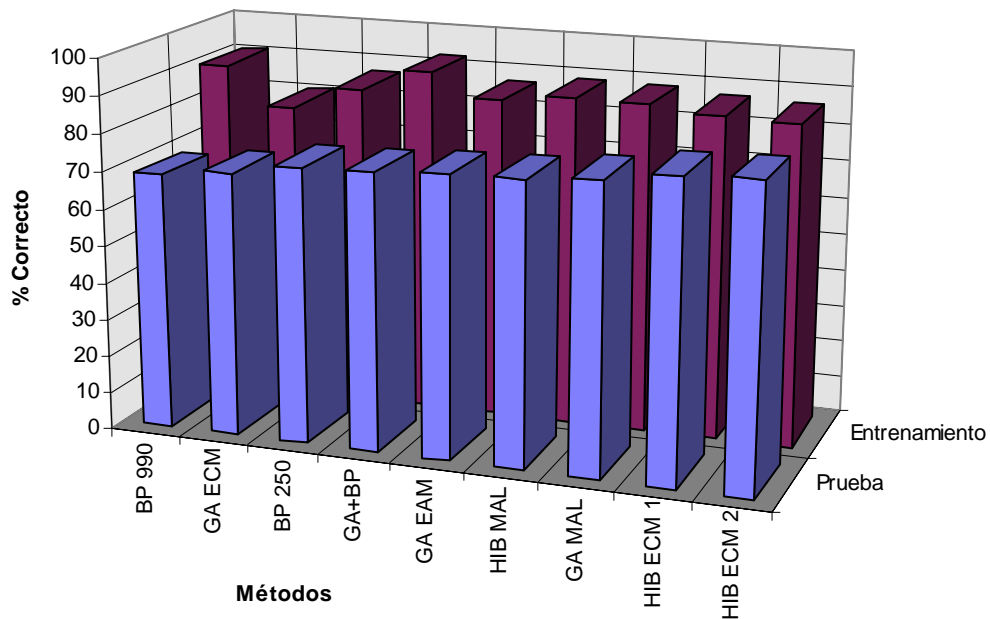


Figura 4: Gráfico comparativo

## Referencias

1. Rich, E., Knight, K, *Artificial Intelligence*, McGraw-Hill, 1991.
2. Sestito, S., Dillon, T.S, *Automated Knowledge Acquisition*, Prentice Hall, 1994.
3. Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., *Classification and Regression Trees*, Wadsworth, Inc, 1984.
5. Kjoelen, A., Thompson, M.J., Scott, E.U., Moss, R.H., Stoecker, W.V., *Performance of A.I. methods in detecting Melanoma*, IEEE Engineering in Medicine and Biology, July/August 1995.
6. R. Lippmann, *An Introduction to Computing with Neural Nets*. IEEE ASSP Mag., Vol 4, 1987.
7. B. Widrow, M. Lehr, *30 Years of Adaptive Neural Networks : Perceptron, Madaline and Backpropagation*, Proceedings of IEEE, Vol. 78, No. 9, pp.1415-1442, Sept. 1990.
8. Hassoun, M.H., *Fundamentals of Artificial Neural Networks*, The MIT Press, 1995.
9. Brent, R.P., *Fast Training Algorithms for Multilayer Neural Nets*, IEEE Transactions on Neural Networks, Vol. 2, No. 3, May 1991.
10. Sethi, I.K., *Decision tree performance enhancement using an artificial neural network implementation*, *Artificial Neural Networks and Statistical Pattern Recognition*, Elsevier Science Publishers, 1991.
11. Goddard J.C., Martinez F.M.; Martinez A.E., Cornejo J.M., Rufiner H.L., Acevedo R.C., *Aprendizaje Maquinal en Medicina: Diabetes, un caso de Estudio*; por aparecer en Revista Mexicana de Ingeniería Biomedica Vol. 17, N°1, 1996.
12. Cios, K., Liu, N., *A Machine Learning Method for Generating a Neural Network Architecture: A Continuous ID3 Algorithm*, IEEE Transactions on Neural Networks, Vol. 3, No. 2, March 1992.
13. Sankar, A., Mammone, R., *Neural Network Trees*, en *Neural Networks*, Academic Press, 1991, Mammone y Zeevi (Eds.).
14. Chi, Z., Yan, H., *ID3-Derived Fuzzy Rules and Optimized Defuzzification for Handwritten Numeral Recognition*, IEEE Transactions on Fuzzy Systems, Vol. 4, No. 1, February 1996.
15. Michie, D., Spiegelhalter, D.J., Taylor, C.C.(Eds.), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, 1994.
16. Hornik, K.M., Stinchcombe, M., White, H., *Multilayer feedforward networks are universal approximators*, *Neural Networks*, 2, 1989.
17. Grandvalet, Y., Canu, S., *Comments on 'Noise Injection into Inputs in Backpropagation Learning'*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 4, April 1995.
18. Montana, D.J., *Neural Network Weight Selection Using Genetic Algorithms*, en *Intelligent Hybrid Systems*, Wiley, 1995, Goonatilake y Khebbal (Eds.).

19. Goddard, J., Lopez ,I., Rufiner, H., *Improving the Classification Accuracy of Cardiac Patients*, enviado al 18<sup>th</sup> Annual Conference of the IEEE Engineering in Medicine and Biology, Amsterdam, 1996.
20. Holland, J.H., *Adaptation in Natural and Artificial Systems*, MIT Press, 1993.
21. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
22. Suzuki, J., *A Markov Chain Analysis on Simple Genetic Algorithms*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 4, April 1995.
23. Michalewicz, Z., *Genetic Algorithms+Data Structures = Evolution Programs*, Springer-Verlag, 1992.
24. *UCI Repository of Machine Learning Databases and Domain Theories*, <ftp://ics.uci.edu/pub/machine-learning-databases>
25. *SUGAL User Manual V2.1*, <http://osiris.sund.ac.uk/ahu/sugal/home.html>
26. Kermani B., White M., Nagle H., *Feature Extraction by Genetic Algorithms for Neural Networks in Breast Cancer Classification*, Proceedings of the 1995 IEEE Engineering in Medicine & Biology, 17th Annual Conference.