

Crituras Virtuales especificadas a través de Redes Neuronales Evolutivas

A.C. Leonardo Corbalán¹, A.C. Mauro Pisano², A.C. Germán Osella Masa²,
Lic. Laura Lanzarini³

***Laboratorio de Investigación y Desarrollo en Informática⁴
Facultad de Informática - Universidad Nacional de La Plata***

Resumen

Las Redes Neuronales Evolutivas proponen combinar las Redes Neuronales convencionales con procedimientos de búsqueda evolutivos a fin de hallar automáticamente una solución eficiente para resolver un problema dado.

El presente trabajo busca evolucionar individuos controlados por redes neuronales que se mueven en un mundo virtual donde se hallan dispersas partículas de comida, que pueden permanecer estáticas o moverse aleatoriamente. El objetivo de cada uno de ellos es alcanzar la partícula más cercana en el menor tiempo posible.

El comportamiento de cada individuo se encuentra representado por una red neuronal cuya definición se ha realizado mediante un algoritmo evolutivo aplicado a cuatro formas de cromosomas diferentes.

Se demuestra que los mejores resultados se obtienen utilizando individuos sin restricciones en las conexiones entre neuronas y con capacidad para evolucionar su función de activación.

Finalmente se muestran y discuten los resultados obtenidos y se plantean líneas de trabajo futuras.

Claves: Redes Neuronales Evolutivas, Aprendizaje, Evolución, Algoritmos Genéticos.

¹ Analista de Computación. Alumno de la Lic.en Informática. UNLP. E-mail: lcor@sinectis.com.ar.

² Analista de Computación. Alumno de la Lic.en Informática. UNLP.

³ Profesor Titular Dedicación Exclusiva. Fac.de Informática. Universidad Nacional de La Plata.
E-mail: laural@info.unlp.edu.ar

⁴ Calle 50 y 115 1er. Piso, (1900) La Plata, Argentina, Tel./Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

1. Introducción

La aplicación eficiente de técnicas de redes neuronales (RN) a la solución de un problema específico implica la selección de tres factores: los elementos de procesamiento, la arquitectura y el algoritmo de aprendizaje. Esto lleva a un modelo estático que permite resolver el problema planteado, cuyo éxito depende fuertemente de la experiencia adquirida en problemas similares [1].

Las Redes Neuronales Evolutivas (RNE) son un caso particular de las redes neuronales (RN) en las cuales la adaptación no sólo se debe al aprendizaje sino también a la evolución [2][7].

La evolución es utilizada con diferentes objetivos, tales como entrenamiento de los pesos [10], diseño de la arquitectura, adaptación de las reglas de aprendizaje, selección de las características de entrada, inicialización de los pesos, etc [3][4].

La característica principal de las RNE es su adaptabilidad a un entorno dinámico. Las dos formas de adaptación de las RNE, evolución y aprendizaje, hacen que su adaptación a un entorno dinámico sea más efectiva y eficiente.

En resumen, las RNE pueden verse como sistemas capaces de cambiar apropiadamente su arquitectura y sus reglas de aprendizaje sin intervención humana [3].

El presente trabajo describe un ambiente que permite evolucionar redes neuronales. Cada individuo de la población está representado por una red neuronal completa. La evolución se aplica tanto sobre la arquitectura como sobre la función de activación de cada neurona.

2. Objetivo

El presente trabajo busca evolucionar individuos, controlados por redes neuronales, que se mueven en un mundo virtual donde se hallan dispersas partículas de comida, que pueden permanecer estáticas o moverse aleatoriamente. El objetivo de cada uno de ellos es alcanzar las partículas en el menor tiempo posible [5][8].

El cromosoma de cada individuo se corresponde con la red neuronal que lo controla. Por lo tanto, es necesario discutir la representación a utilizar a fin de obtener los mejores resultados.

3. Tipos de individuos utilizados

Se han desarrollado cuatro tipos de individuos. Los más sencillos poseen una red neuronal de tres capas y corresponden a los tipos **TindividuoBP** y **TindividuoBPMultiSigmoide**. Los más complejos poseen una arquitectura de red prácticamente sin restricciones (grafos genéricos) y corresponden a los tipos **Tindividuo** y **TindividuoMultiSigmoide**.

3.1. Individuo TindividuoBP

Descripción de la arquitectura

La arquitectura de la RN que representa a este individuo está basada en la red Backpropagation [1][9], es decir que, existe una única capa de neuronas ocultas que sólo reciben información de las neuronas de entrada y únicamente envían información a las neuronas de salida.

La evolución de la estructura está dada por la disposición de las conexiones.

La cantidad de neuronas en las capas de entrada y salida están determinadas por el problema. En cuanto a la capa oculta, la misma posee un máximo de neuronas pero el algoritmo evolutivo podría conseguir redes con menor cantidad ya que las neuronas de la capa oculta podrían no estar conectadas. Todas las neuronas, excepto las de entrada, poseen una conexión de tendencia.

Codificación del cromosoma

El cromosoma concentra la información genética de una red neuronal completa y la unidad mínima de información es el bit (un alelo).

Ya que las neuronas de entrada no poseen conexiones de entrada, y las de salida no poseen conexiones de salida basta con codificar las neuronas ocultas y las de salida de la siguiente manera:

- Por cada neurona oculta se codifica el peso de la conexión de tendencia más las conexiones de entrada provenientes de cada una de las neuronas de entrada.
- Por cada neurona de salida se codifica el peso de la conexión de tendencia más las conexiones de entrada provenientes de cada una de las neuronas ocultas.
- Las conexiones se codifican utilizando dos valores: un “bit de existencia” que indica la presencia o no de la conexión y el peso de dicha conexión.

Esquema de un cromosoma BP para 6 neuronas de entrada, 10 ocultas y 3 de salida.

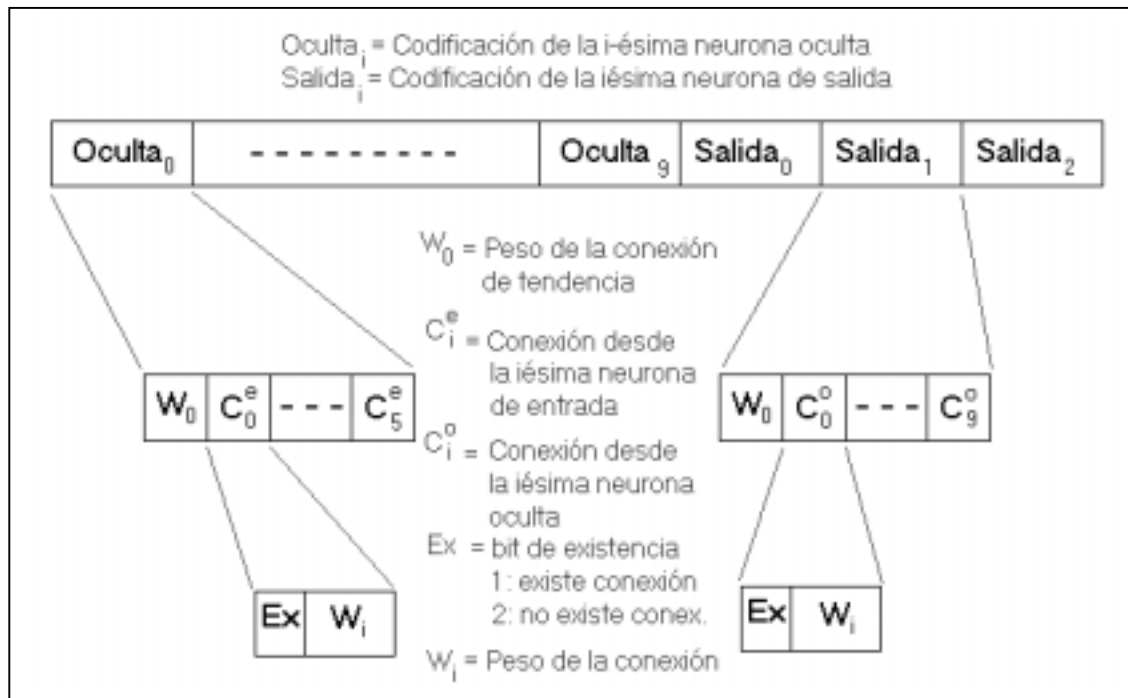


Fig.1

3.2. Individuo TindividuoBPMultiSigmoide

Posee una estructura casi idéntica a la de un TindividuoBP. La única diferencia radica en la función sigmoide que se aplica tanto en la evaluación de las neuronas ocultas como en las de salida. Para un TindividuoBPMultisigmoide dicha función no es única, sino que cada neurona tiene asociada una sigmoide entre cuatro tipos distintos.

Por lo tanto, la codificación del cromosoma es similar al TindividuoBP incorporando dos bits por neurona para identificar uno de los cuatro tipos de sigmoide posibles.

Los tipos de sigmoide utilizados son los siguientes:

- Tipo 1) $f(x) = 2/(1+\exp(-0.5*x))-1$
- Tipo 2) $f(x) = 2/(1+\exp(-x))-1$
- Tipo 3) $f(x) = 2/(1+\exp(-1.5*x))-1$
- Tipo 4) $f(x) = 2/(1+\exp(-2*x))-1$

3.3. Individuo Tindividuo

Descripción de la arquitectura

Considera como red neuronal válida a un grafo dirigido dónde los nodos representan neuronas y los arcos sus conexiones, con las siguientes restricciones:

- Deben existir al menos una neurona de entrada y al menos una neurona de salida.
- Las neuronas de entrada no pueden tener conexiones de entrada.

La existencia de ciclos en el grafo brindan la posibilidad que la red “recuerde” información pasada. Todas las neuronas, excepto las de entrada, poseen una conexión de tendencia.

Codificación del cromosoma

- Por cada neurona se codifica una conexión hacia cada una de las neuronas que no son de entrada.
- Por cada neurona que no sea de entrada se codifica el peso de la conexión de tendencia.
- Las conexiones se codifican utilizando dos valores: un “bit de existencia” que indica la presencia o no de la conexión y el peso de dicha conexión.

La figura 2 ejemplifica la codificación del cromosoma para una red con: **p** neuronas de entrada **q** neuronas ocultas y **r** neuronas de salida

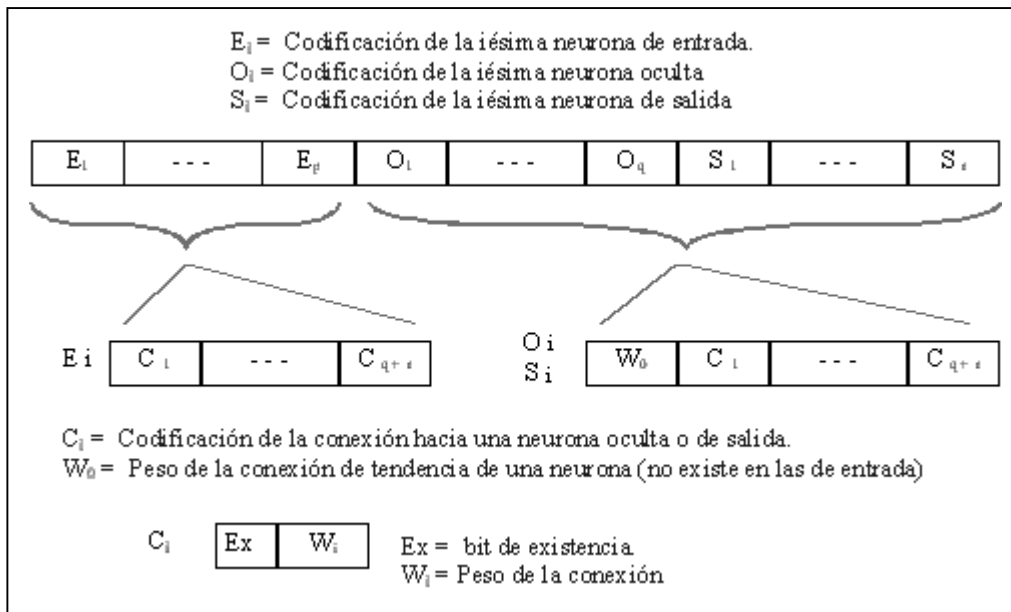


Fig.2

3.4. Individuo TindividuoMultiSigmoide

La única diferencia entre un Tindividuo y un TindividuoMultiSigmoide es que en éste último las neuronas ocultas y de salida tienen asociada una sigmoide entre cuatro tipos distintos. Estos tipos de sigmoide son los mismos que se presentan en el TindividuoBPMultiSigmoide.

La codificación del cromosoma es como la de Tindividuo salvo que para las neuronas ocultas y de salida, se agregan dos bits para identificar el tipo de sigmoide. La codificación de las neuronas de entrada es idéntica a la de Tindividuo.

La codificación de las neuronas ocultas y de salida es la siguiente:

TS	W ₀	Conexiones
----	----------------	------------

donde TS=2 bits. Con este campo se codifica el tipo de sigmoide que puede ser 1,2,3 ó 4.
W₀ y Conexiones idéntico a Tindividuo

4. Evolución de individuos

Dentro del mundo virtual se encuentran dispersas partículas de comida que pueden permanecer estáticas o moverse aleatoriamente.

Dado que el objetivo es lograr que cada individuo se aproxime a la partícula más cercana, su fitness estará determinado por la distancia negativa a la comida más cercana. De esta forma, el mayor valor de fitness representará al mejor individuo.

Los cuatro tipos de individuos fueron evolucionados utilizando un algoritmo genético elitista con tamaño de población fija.

El pseudocódigo del algoritmo utilizado es el siguiente:

Generar al azar una población inicial de P individuos y evaluar los fitness de cada uno de ellos.

Realizar N generaciones de la siguiente forma:

- *Obtener una población de hijos de igual tamaño que la de padres realizando la selección de pares por el método de la ruleta y aplicando crossover de dos puntos y mutación según corresponda. Cada individuo llevará asociado su fitness.*
- *Reemplazar la población de padres por la de hijos utilizando un criterio elitista.*
- *Conservar el mejor individuo de esta generación para un análisis posterior.*

donde P y N son parámetros.

5. Análisis de la evolución de cada tipo de individuo

Dado un tipo de individuo, es necesario analizar la mejora de su capacidad para acercarse a la comida más cercana a medida que avanzan las generaciones. Esto permite afirmar que los individuos evolucionan favorablemente.

Para ello, es imprescindible uniformar la manera de obtener los fitness correspondientes. Nótese que la posición de las partículas de comida puede variar de una generación a otra. Esto impide comparar los fitness aun en generaciones consecutivas.

Como forma de resolver el problema, se evolucionan 100 poblaciones de cada tipo de individuo, realizando 75 generaciones en cada evolución. De cada generación sólo se guarda el mejor representante. Luego de esto, cada generación de un mismo tipo de individuo contará con 100 redes neuronales, que a través de su fitness promedio, representan el efecto del algoritmo genético sobre la evolución del cromosoma. Dicho en otras palabras, para cada tipo de individuo se dispone de 100 redes neuronales representativas de cada generación.

El fitness de cada red, que en este caso mide la capacidad evolucionada de cada individuo para acercarse a la comida más próxima lo más rápido posible, se mide ubicando la comida en 8 puntos cardinales fijos como indica la Fig. 3.

Los 7500 individuos parten del centro realizando, cada uno de ellos, 1000 intentos por llegar a la partícula ubicada en la posición 1. Luego retornan al centro y repiten este proceso con la partícula ubicada en la posición 2. Así siguiendo, hasta cubrir todas las posiciones.

Finalmente, el fitness de cada generación se calcula como el promedio de los fitness de sus 100 individuos obtenidos de los 8000 intentos anteriores.

Con la obtención de esta medida es ahora posible comparar el efecto del algoritmo genético sobre cada tipo de individuo.

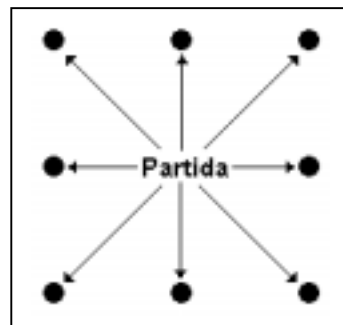


Fig.3

6. Resultados Obtenidos

Los resultados han sido medidos sobre diferentes lotes de individuos. Cada lote está caracterizado por la cantidad de neuronas utilizadas tanto en la capa oculta como en la de salida.

En todos los casos los índices 1, 2, 3 y 4 corresponden a los tipos de individuo Tindividuo, TindividuoMultiSigmoide, TindividuoBP y TindividuoBPMultiSigmoide respectivamente.

6.1 Evolución de cada tipo de individuo

Se comenzará el análisis trabajando sobre individuos con dos neuronas de entrada, 4 ocultas y dos neuronas de salida.

Los pertenecientes al lote A (fig.4) reciben como entradas las distancias exactas a la comida más próxima, tanto en el eje x como en el eje y. En cambio, los individuos del lote B (fig.5) reciben como entradas valores 0, 1 o -1 indicando la dirección de la comida más próxima tanto en el eje x como en el eje y. En ambos lotes, las neuronas de salidas representan las fuerzas que el individuo deberá aplicar en cada dirección para aproximarse al objetivo.

Las figuras 4 y 5 muestran que los individuos genéricos, los índice 1 y 2, son superiores a los individuos basados en BackPropagation, de índices 3 y 4.

Además, en la fig.5 se observa que la característica Multisigmoide (índices 2 y 4) mejora el fitness tanto en los individuos del tipo BP como en los de tipo genérico. Esto último no se observa en la fig.4 entre A1 y A2 debido a que el Lote A recibe mejor información de entrada compensando de esta forma el efecto MultiSigmoide.

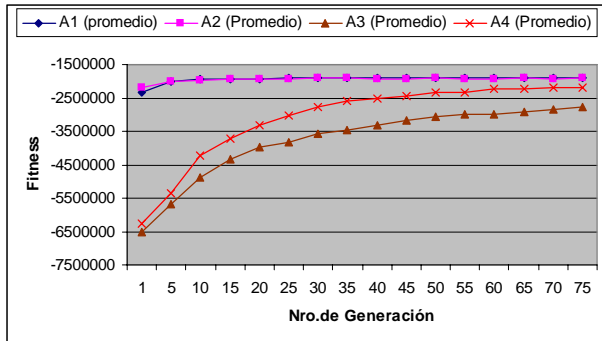


Fig.4

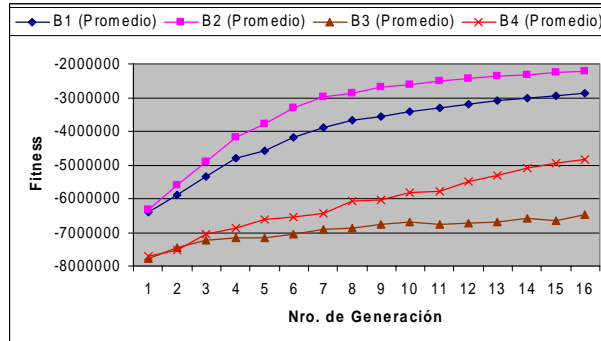


Fig.5

6.2 Cambio en el cromosoma del individuo genérico

La población del lote C es similar a la del B salvo que incorpora una neurona de salida que permite obtener un valor constante que se utiliza para multiplicar a las fuerzas que el individuo debe realizar para llegar a la comida.

Es decir, en el lote B la salud es de la forma (F_x, F_y) y en C es (F_x, F_y, Cte) .

La figura 6 muestra la relación entre los individuos multisigmoide. Puede verse que los genéricos tienen mejor desempeño que los Backpropagation. Pese a haber realizado pruebas con un mayor número de generaciones no se aprecia un cambio significativo en este aspecto.

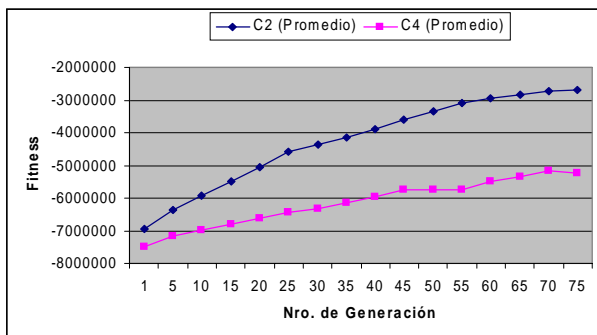


Fig.6

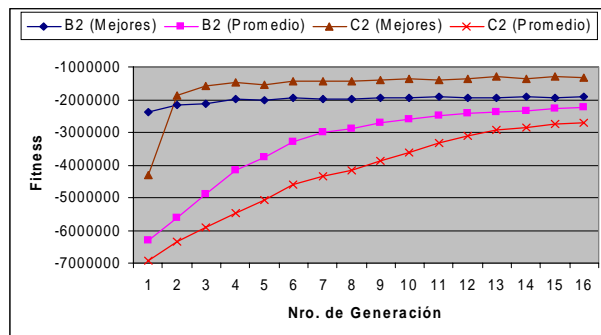


Fig.7

Por otro lado, si se comparan los individuos B2 y C2 (fig.7) puede verse que los mejores individuos están en C2. Es importante recordar que el fitness del mejor de cada generación es obtenido promediando los fitness de las 100 simulaciones. Sin embargo, si se considera el fitness promedio de todos los individuos de cada generación, puede observarse que B2 supera a C2.

6.3 Representando las fuerzas de salida con diferente intensidad

La población F está formada por individuos con las siguientes características: $RN=(2,4,3)$, $entradas=(\text{signo}(\Delta x), \text{signo}(\Delta y))$ y $salida=(F_x, F_y, Cte \text{ que multiplica a } F_x, Cte \text{ que multiplica a } F_y)$. Es decir, ahora se utilizan valores constantes específicos para cada una de las fuerzas de salida.

Las figuras 8, 9, 10 y 11 muestran la comparación entre los individuos genéricos de los lotes B, C y F. Se recuerda que la única diferencia entre los grupos es la cantidad de neuronas de salida en las redes neuronales.

Salida del grupo B: (F_x, F_y)

Salida del grupo C: $(F_x, F_y, Cte \text{ que multiplica a ambas fuerzas})$

Salida del grupo F: $(F_x, F_y, Cte \text{ que multiplica a } F_x, Cte \text{ que multiplica a } F_y)$

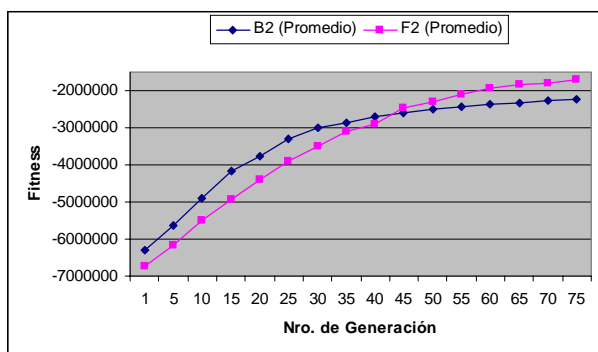


Fig.8

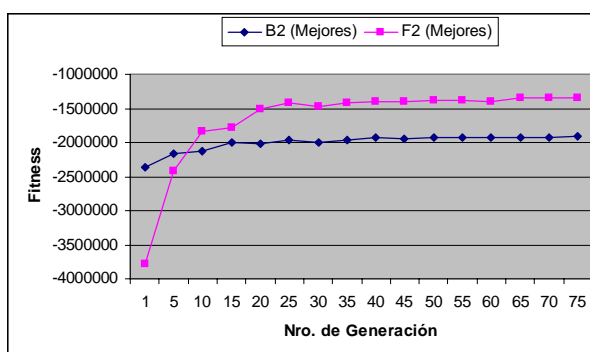


Fig.9

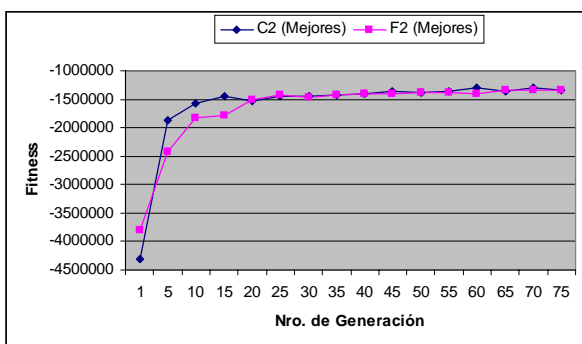


Fig.10

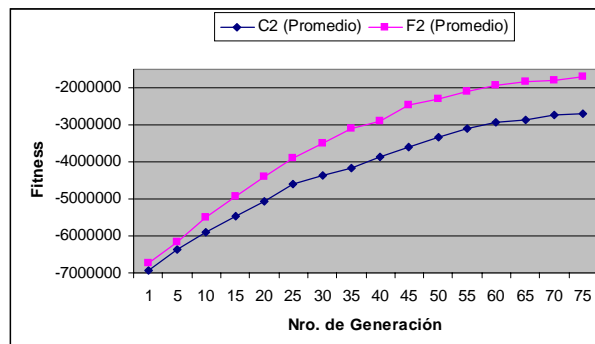


Fig.11

En la comparación de los individuos genéricos del lote F con los del B, se observa que el agregado de las dos salidas constantes que multiplican a F_x y F_y respectivamente, mejora el fitness tanto para el individuo promedio (fig.8) como para el caso del mejor individuo (fig.9).

Por otro lado, si se compara F2 con C2 (fig.10) puede verse que los mejores individuos en ambas poblaciones son equivalentes. Pero el análisis de las poblaciones promedio (fig.11) muestra que existe una mayor diversidad genética en C2.

Comparación entre los distintos tipos de individuos

Las figuras 12 y 13 resumen la evolución de los lotes analizados previamente. La barra de mayor tamaño representa el mejor fitness. De esta forma, hacia la izquierda del gráfico aparecen las pruebas que generaron individuos de mayor fitness.

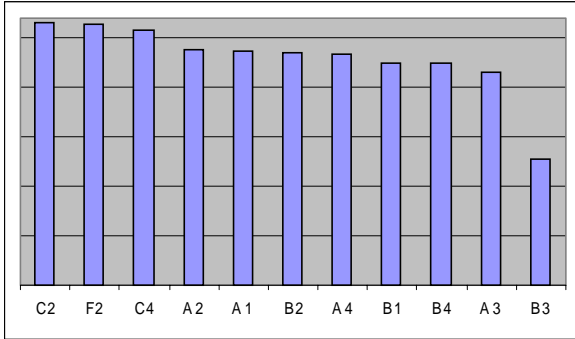


Fig.12 Mejores individuos

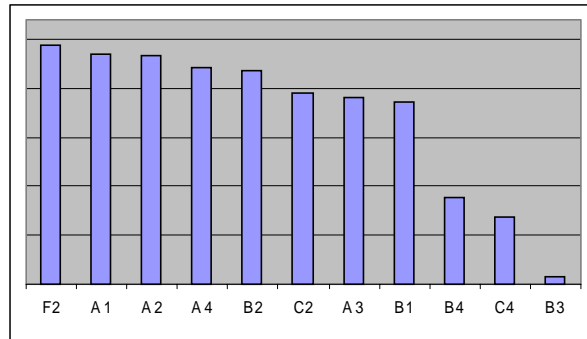


Fig.13. Individuos Promedio

La figura 12 muestra los mejores individuos de cada lote, mientras que la figura 13 utiliza el individuo promedio.

En general, se observa en ambas figuras que los subíndices “2” se recuestan a la izquierda y los subíndices “3” hacia la derecha. Esto pone en evidencia que los individuos genéricos con sigmoides “ajustables” (2) son quienes mejor se comportan. En el otro extremo los individuos tipo backpropagation con sigmoides fijas (3) son los de peor performance.

Nótese que las soluciones obtenidas del lote C son tan buenas como las del F pero conservan una muy buena diversidad genética.

7. Conclusiones y líneas de trabajo futuras

Las pruebas anteriores han sido realizadas con 75 generaciones por cada una de las 100 simulaciones de cada tipo de individuo a analizar. El parámetro 75 fue estimado luego de varias mediciones sobre las poblaciones. La figura 14 permite comparar los individuos de F2 con los de una población EXTRA que realiza 500 generaciones en cada simulación, corroborando lo dicho anteriormente.

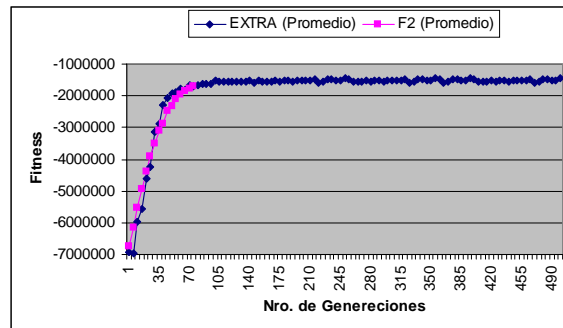


Fig.14

En general se ha observado que luego de 70 generaciones el aumento del número de individuos en la población no representa una ventaja significativa

A partir de los resultados expuestos puede afirmarse que los individuos genéricos evolucionan mejor que los basados en Backpropagation. Además, la característica multisigmoide mejora aun más estos resultados.

Sin embargo, aun quedan varios aspectos por cubrir:

En cuanto a las evaluaciones

- El diseño de un campo de pruebas más complejo, por ejemplo con 64 puntos cardinales en lugar de 8, permitiría evaluar con mayor precisión el comportamiento de los individuos acercándose a partículas de comida que se hallan a distintos ángulos en relación a ellos.
- Realizar evoluciones de muchas generaciones (p.ej.500) para detectar si existe algún cambio de consideración más allá de la generación 75. En el grupo de prueba EXTRA, se observó que en algunas generaciones hay una caída y recuperación de fitness más allá de la generación 400. Es más, el mayor fitness obtenido aparece en la evolución 6 inmediatamente luego de la recuperación de una caída de la curva.
- Intentar encontrar características comunes en las estructuras de las RN exitosas

En cuanto al algoritmo genético

- Modificar el fitness para que premie las estructuras más simples, de esta manera se generarán redes con menos neuronas y conexiones más fáciles de comparar entre sí.

En cuanto a los problemas a resolver

- Modificar la información con que se alimenta las redes neuronales para que pueda ver más de una comida, todas o quizá algunas en un radio determinado. Esto agrega el problema de reconocer a la más cercana, situación que en el presente trabajo se da por resuelto.
- Agregar a los problemas un grado de dificultad. Por ejemplo, sería interesante que hubiese algún otro tipo de partículas además de la comida, tal como partículas que deberían evitarse (campo minado) intentando llegar a la comida.
- Una vez resuelto el problema del campo minado, una nueva e interesante complicación sería evolucionar individuos capaces de aprender un recorrido (salir de un laberinto)

Toda la documentación de este trabajo junto con el ambiente desarrollado ad-hoc para realizar la simulación y medidas aquí mencionadas, se encuentran disponibles en el LIDI (Lab.de Investigación y Desarrollo en Informática).

Referencias

- [1] Freeman y Skapura. "Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación". Addison-Wesley. 1993
- [2] Xin Yao. "Evolving Artificial Neural Networks". PIEEE: Proceedings of the IEEE, vol 87, Setiembre 1999
- [3] Balakrishnan, K. and Honavar, V. "Evolutionary Design of Neural Architectures: A Preliminary Taxonomy and Guide to Literature". Department of Computer Science, Iowa State University, Ames, Iowa. 1995"
- [4] G. F. Miller and P. M. Todd and S. U. Hedge". "Designing Neural Networks Using Genetic Algorithms". Proceedings of the Third International Conference on Genetic

Algorithms, George Mason University, Arlington. pp 379-384. 1989

- [5] Karl Sims "Evolving Virtual Creatures". Published in: Computer Graphics, Annual Conference Series, (SIGGRAPH '94 Proceedings) July 1994, pp.15-22
- [6] David E. Goldberg. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley Publishing Company, Inc. 1989
- [7] X. Yao and Y. Liu. "A New Evolutionary System for Evolving Artificial Neural Networks". IEEE Transactions on Neural Networks, Vol 8, nro. 3, pp 694-713, 1997
- [8] Pablo Funes and Jordan B. Pollack. "Computer Evolution of Buildable Objects".Fourth European Conf. on Artificial Life". MIT Press. Cambridge, MA. P. Husbands and I. Harvey. Pages 358-367. 1997
- [9] Robert Hecht-Nielsen. "Neurocomputing". Addison-Wesley Publishing Company.1990
- [10] Plagianakos, V.P.; Vrahatis, M.N. "Training neural networks with threshold activation functions and constrained integer weights ". Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on , Volume: 5 , 2000. Page(s): 161 -166 vol.5