

## Redes Neuronales Cooperativas para el Control de un Robot Autónomo

Oscar Luis Ruíz Bernal, Saúl Martínez Díaz

*División de Estudios de Posgrado e Investigación, Instituto Tecnológico de La Paz*

[rboscar7@yahoo.com](mailto:rboscar7@yahoo.com), [saulmd@itlp.edu.mx](mailto:saulmd@itlp.edu.mx)

### Resumen

*En este trabajo se propone el uso de redes neuronales cooperativas para controlar un robot autónomo. El entrenamiento de las redes se llevó a cabo utilizando un algoritmo genético. Además se realizó una comparación entre la estructura propuesta y una estructura de red neuronal única. Los resultados obtenidos por medio de simulaciones muestran un mejor desempeño de la estructura cooperativa con respecto a la estructura monolítica.*

### 1. Introducción

En los últimos años se ha incrementado considerablemente la aplicación de redes neuronales artificiales (RNA) en la solución de problemas diversos. En gran medida esto se debe a su capacidad de aprender, generalizar y organizar datos [1]. Esta idea resulta particularmente atractiva en el área de la robótica, donde se trata de emular un comportamiento inteligente en el robot para realizar una tarea determinada.

Uno de los aspectos aún en estudio de las RNA's es la estructura que debe tener la red. En [2] se utilizó una RNA única para que un robot seleccionara una acción entre un conjunto de acciones posibles. Bajo este esquema se requiere una red grande de la cual depende totalmente la tarea a realizar. Otra posibilidad que se ha propuesto es el uso de varias redes que trabajan de manera colaborativa. De este modo una tarea se divide en sub-tareas más simples, cada una de las cuales se distribuye entre cada red [3]-[7]. En [3] se propuso el alineamiento de imágenes utilizando redes neuronales cooperativas. En [4] se utilizó este enfoque modular para prevenir intrusiones en redes de computadoras y dispositivos móviles. En [5] se utilizó este tipo de red modular para clasificar datos obtenidos por sensores remotos. En [6] se propuso una estructura de sub-agentes simples para controlar un robot autónomo mediante el uso de múltiples RNA's. Además, en [7]

se presenta una aplicación utilizando una arquitectura de múltiples redes cooperativas para controlar un robot dirigible. En este último caso los resultados muestran que dicha arquitectura es más robusta a fallas del robot que una arquitectura monolítica.

En este trabajo se propone una arquitectura de RNA's cooperativas para resolver una tarea determinada, utilizando un robot autónomo. La tarea consiste en encontrar un objeto (una pelota), buscar un contenedor y colocar el objeto. Además se presenta una comparación de la capacidad de generalización de la arquitectura propuesta con respecto a una arquitectura monolítica. En ambos casos se utilizó un algoritmo genético (AG) para entrenar las redes. Debido a sus características, los AG's y las RNA's han tendido a fusionarse en diversas aplicaciones de inteligencia artificial [8]. El entrenamiento se efectuó de forma simultánea en varias computadoras personales, para incrementar su velocidad.

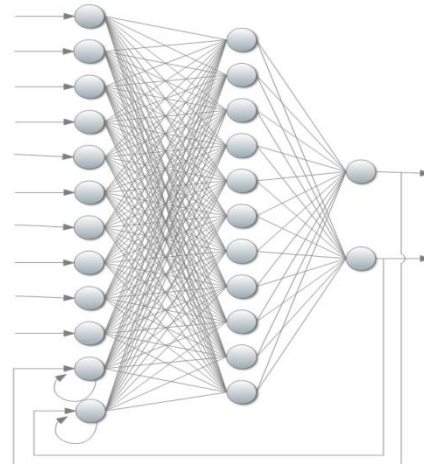


Figura 1. Estructura de red neuronal monolítica.

### 2. Redes neuronales

La arquitectura de las RNA's utilizadas se diseñó utilizando retroalimentación (red recurrente). Este tipo de redes han demostrado ser más poderosas y tener una

mayor estabilidad que las redes no recurrentes en aplicaciones de planeación de trayectorias y control [9]. La estructura de red neuronal monolítica se muestra en la figura 1. Por otro lado las redes cooperativas tienen la misma estructura, la cual se muestra en la figura 2.

En la red neuronal de estructura monolítica se definen tres capas de neuronas una de ellas oculta, la capa de entrada cuenta con doce neuronas, nueve reciben información de los sensores de distancia, una recibe cero o uno al haber capturado o no la pelota, y las dos siguientes reciben retroalimentación de la capa de salida. La capa oculta cuenta con 11 neuronas sin retroalimentación de la capa de salida o de sí misma. Finalmente se tienen dos neuronas de salida que definen el movimiento de cada rueda del vehículo y cada neurona retroalimenta a una neurona de entrada diferente.

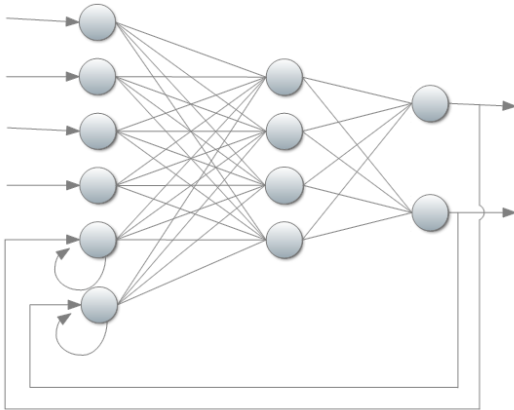


Figura 2. Estructura de red neuronal monolítica.

El modelo de redes cooperativas utiliza tres redes neuronales con la misma estructura. La primera red se configura para evitar chocar, la segunda para detectar de manera lateral los objetos, y la tercera para detectar de manera frontal los objetos.

En cada red cooperativa se definen tres capas de neuronas una de ellas oculta, la capa de entrada cuenta con seis neuronas, cada red recibe diferente información de entrada. La capa oculta cuenta con cuatro neuronas sin retroalimentación de la capa de salida o de sí misma. Finalmente se tienen dos neuronas de salida que definen el movimiento de cada rueda del vehículo y cada neurona retroalimenta a una neurona de entrada diferente.

El valor final de salida que define la velocidad y la dirección del vehículo es la sumatoria de las salidas de las tres redes neuronales. Esto es, la neurona de salida 1 de cada red neuronal se suma para alimentar a la

rueda izquierda, la neurona de salida 2 de cada red neuronal se suma para alimentar a la rueda derecha.

Se utiliza retroalimentación a sí misma para que cada neurona establezca un nivel de “estados”, donde una parte de la entrada sea un historial de los comportamientos anteriores. Con la finalidad de incrementar la influencia de los comportamientos anteriores, la entrada de estas neuronas se calcula por:

$$E = (S * 0.2) + (E * 0.8) \quad (1)$$

Donde E es el valor de entrada y S es la salida que se retroalimenta. El parámetro de umbral no se utiliza. El resto de las neuronas se definen con la sumatoria de todas las entradas y en cada neurona se utiliza como función de activación la siguiente función, la cual permite utilizar entradas continuas y discretas con un rango de salida de [0, 1].

$$y = \frac{1}{1 + e^{-x}} \quad (2)$$

### 3. Algoritmo Genético

De acuerdo con Zbigniew Michalewicz [10] un algoritmo genético está compuesto por 5 elementos:

- Un individuo como una representación genética de la solución del problema, el cual es evaluado para seleccionar las mejores soluciones,
- Un camino para crear una población inicial de soluciones,
- Una función de evaluación de la calidad de las soluciones en términos de su eficiencia,
- Operadores genéticos tal que intercambien componentes genéticos en los hijos durante la reproducción y
- Un grupo de valores adecuados para los parámetros del algoritmo genético.

Los AG’s se utilizan principalmente porque pueden trabajar sobre varias soluciones de forma simultánea, lo cual facilita su implementación en paralelo. Además, al utilizarse en problemas de optimización son menos susceptibles de quedar atrapados en óptimos locales, comparados con los métodos de búsqueda local. Los parámetros utilizados para el AG fueron:

Población = 100

Generaciones = 50

Tamaño del torneo = 3

Probabilidad de cruce = 70%

Probabilidad de mutación = 10%

Estos parámetros fueron obtenidos de manera experimental, buscando aquellos que dieran mejores resultados. Durante el cruce se hace una selección aleatoria del número de cromosomas a intercambiar. El porcentaje de mutación disminuye con cada

generación. Se utiliza el elitismo para conservar el mejor de cada generación. El operador de selección por torneo presenta la característica de preservar sólo a los mejores individuos a diferencia del método de la ruleta donde existe la posibilidad de que el peor calificado de los individuos pase a la siguiente generación. El número de individuos es de tres ya que da mayor seguridad sobre la supervivencia de los mejor calificados. Esta decisión va acorde al uso del elitismo, el cual permite que el mejor de los individuos se conserve para la siguiente generación, de modo que en la última generación se tenga el mejor individuo hasta ese momento.

En el caso particular de las redes independientes se toman como un sólo individuo por lo que evolucionan en su conjunto, esto es, se suma el valor de eficiencia de las tres redes para estimar su valor al momento del torneo y se seleccionan las tres para conformar el siguiente individuo. Al momento del cruce se intercambian cromosomas entre las redes del mismo tipo. La función de aptitud le da puntos por diferentes conceptos:

- El máximo valor se obtiene al depositar la pelota en el contenedor (10,000).
- Al encontrar la pelota (1,000).
- Al detectar la pelota o el contenedor por los sensores frontales se agrega 0.1 si la diferencia de velocidad en ambas ruedas es menor a 0.05 (en un rango de -3 a 3), en caso contrario se resta 0.1.
- Al detectar la pelota o el contenedor por los sensores laterales se agrega 0.1 si la diferencia de velocidades de las ruedas es menor a 0.1 a favor del sentido donde fue detectado el objeto, en caso contrario se resta 0.1.
- Cuando hay un choque con la pared se le resta 0.1. En este caso el vehículo retrocede durante 10 tiempos de manera automática.

Se obtienen valores muy grandes al encontrar la pelota y posteriormente el contenedor ya que la meta es encontrar primero la pelota y luego el contenedor. El objetivo es que las recompensas por detectar los objetos sean menos significativas que el hecho de encontrarlos. El valor asignado al contenedor no se da si no se ha encontrado la pelota; esto es, si la pelota no se ha encontrado y se encuentra primero el contenedor no genera valor y la búsqueda continúa. Sólo cuando se encuentra la pelota y luego el contenedor se detiene la prueba y se suma la recompensa por el éxito.

Al detectar la pelota de frente se toma en cuenta la dirección del vehículo, que está determinada por la velocidad de cada rueda. Dado que es muy poca la probabilidad de que la diferencia de velocidades sea cero, se toma un margen de 0.05 para considerar que el vehículo va de frente y asignar la recompensa. Dicho

margen también se aplica al detectar la pelota por un costado y asignar la recompensa, sólo si el vehículo da vuelta en el sentido que se detecta el objeto.

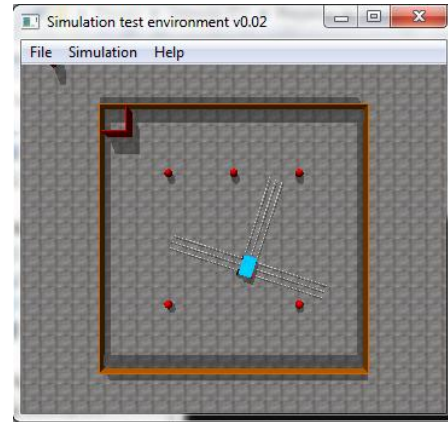


Figura 3. Configuración con 5 pelotas.

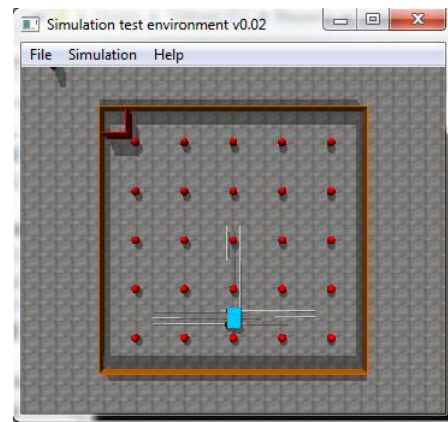


Figura 4. Configuración con 25 pelotas.

De cada entrenamiento se toma al mejor de los individuos y se le aplicó una evaluación final de eficiencia que consta de 40,000 pasos. En esta prueba sólo se recompensó al encontrar la pelota después del contenedor. Esta evaluación final para ambas estructuras consta de dos módulos de prueba de eficiencia, uno utilizando posiciones aleatorias y otro utilizando un conjunto de posiciones predeterminadas. En el primer módulo de prueba cada individuo realiza 21 pruebas con una pelota en posición aleatoria. En el segundo módulo de prueba cada individuo realiza 25 pruebas con una pelota en posición predeterminada, como se muestra en la figura 4.

Uno de los factores más importantes en los resultados, es el porcentaje de aciertos obtenido por cada estructura de red. La tabla 1 presenta los

resultados obtenidos por ambas estructuras en las pruebas con la pelota en posición aleatoria. Como se puede observar, la configuración propuesta de redes cooperativas tuvo mejores resultados que la red única. Por otro lado, en la tabla 2 se presentan los resultados de las pruebas con las pelotas en posiciones fijas. Una vez más la configuración de múltiples redes presentó mayor eficiencia.

**Tabla 1.** Resultados de experimentos en posiciones aleatorias de la pelota.

Estructura de Red	% Aciertos
Única	81.0
Cooperativas	85.7

**Tabla 2.** Resultados de experimentos en posiciones fijas de la pelota.

Estructura de Red	% Aciertos
Única	68.0
Cooperativas	76.0

La figura 5 muestra el porcentaje de pruebas que superan cada nivel de efectividad. Como se observa, ambas estructuras siguen una misma tendencia con una diferencia en el porcentaje de aproximadamente el 6% a favor de la red cooperativa.

En las figuras 6 y 7 se ven las 25 posiciones iniciales de la pelota para la red única y las redes cooperativas, respectivamente. En dichas figuras se indica el número de veces que se logró el objetivo de depositar la pelota en el contenedor en cada posición. La posición que logra un mayor número de éxitos es la que se encuentra más cercana al contenedor, de la misma forma el lado izquierdo obtiene el mayor número de éxitos en ambas estructuras, sin embargo en la red única el siguiente lado que obtiene mejores resultados en la parte superior mientras que la red cooperativa es la parte inferior.

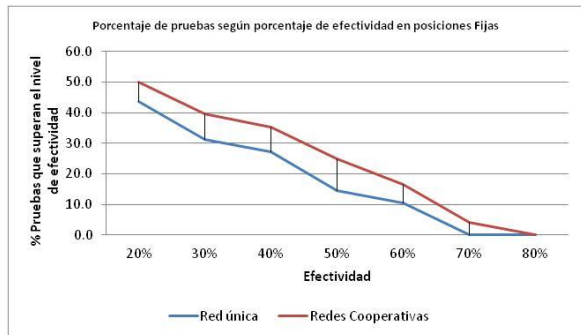


Figura 5. Porcentaje de efectividad de las pruebas.

Como era de esperarse las posiciones que obtuvieron un menor número de éxitos son las posiciones centrales, esto es debido a que en el entrenamiento no se utilizaron pelotas ubicadas en esta posición. Sin embargo, las redes cooperativas fueron más eficientes en encontrar las pelotas ubicadas en el centro, a pesar de no haber sido entrenadas para ello. En la figura 8 se muestra el nivel de efectividad en función de la distancia euclidiana de cada ubicación de la pelota con respecto al centro del área de búsqueda. El punto central es el más alejado de las posiciones de las pelotas usadas en el entrenamiento, en consecuencia es el que tiene menor efectividad en ambos casos; sin embargo, la red cooperativa muestra mejores resultados en los puntos más alejados. Esto significa que, en este caso, la estructura cooperativa tuvo una mejor capacidad de generalizar que la estructura única.

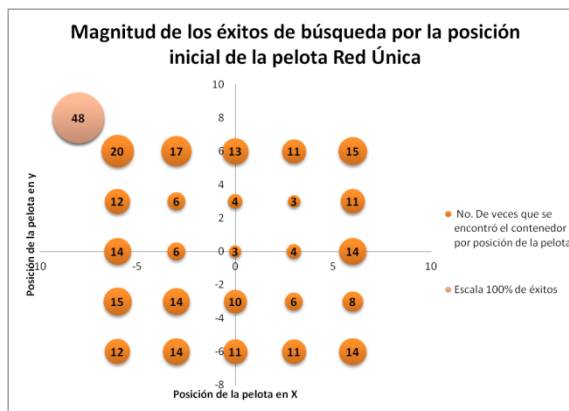


Figura 6. Resultados con red única.

## 5. Conclusiones

En este trabajo se propuso el uso de redes cooperativas en robots, para realizar una tarea específica. Se efectuaron simulaciones por computadora para comparar la estructura propuesta contra una estructura de red única. Los resultados muestran que la red cooperativa alcanzó el objetivo un mayor número de veces que la red única y tuvo una mejor capacidad de generalización. Como trabajo futuro se propone modificar los parámetros de entrenamiento y la función de aptitud, para buscar mejores resultados.

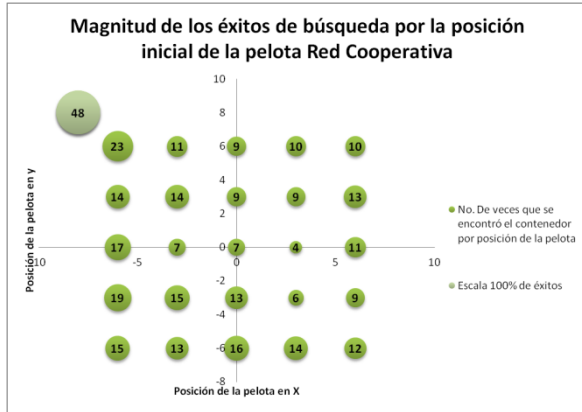


Figura 7. Resultados con redes cooperativas.

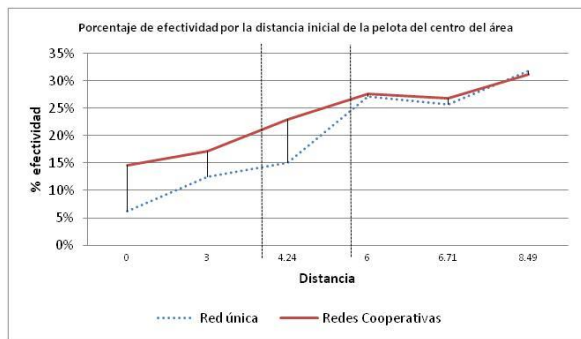


Figura 8. Efectividad según distancia.

## 6. Referencias

[1] B. Krose and P. V. Smagt. *An Introduction to Neural Network*, The University of Amsterdam, Netherland, 1996.

[2] F. Montes-Gonzalez, C. A. Ochoa-Ortiz-Zezzatti, L. F. Marín-Urías and J. Sánchez-Aguilar, “Un Enfoque Híbrido en el Desarrollo de Robótica Basada en el Comportamiento”, *Computación y Sistemas*, 2010, pp. 385-397.

[3] C. Y. Hsu, Y. C. Cheng and S. F. Lin, “Precise Image Alignment Using Cooperative Neural-Fuzzy Networks with Association Rule Mining–Based Evolutionary Learning Algorithm”, *Opt. Eng.*, 2012, pp. 027006-1-15.

[4] J. Cannady, “An Adaptive Neural Swarm Approach for Intrusion Defense in Ad Hoc Networks”, In *Proceedings of SPIE*, 2011, pp. 80590P-1-8.

[5] P. Blonda, V. La-Forgia, G. Pasquariello and G. Satalino, “Feature Extraction and Pattern Classification of Remote Sensing Data by a Modular Neural System”, *Opt. Eng.*, 1996, pp. 536–542.

[6] R. A. Tellez y C. Angulo. “Generando un Agente Robótico Autónomo a Partir de la Evolución de sub-Agentes Simples Cooperativos”, In *Workshop En Agentes Fisicos*, 2004, pp. 113-118.

[7] H. Kawamura, H. Iizuka, T. Takaya and A. Ohuchi, “Cooperative Control of Multiple Neural Networks for an Indoor Blimp Robot”, *Artif Life Robotics*, 2009, pp. 504–507.

[8] L. C. Jain and N.M. Martin, *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*, CRC Press, Australia, 1998.

[9] S. B. Unadkat, M. M. Ciocoiu and L. R. Medsker, *Recurrent Neural Network Design and applications*, CRC Press, Boca Raton, London, New York, Washington, D.C. 2001.

[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1999.