

Aplicaciones de Redes Neuronales a Problemas Teóricos y de Control

Carlos Kavka, Patricia Roggero y Javier Apolloni

LIDIC

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950

D5700HHW - San Luis - Argentina

Tel: 02652-420823 - Fax: 02652-430224

e-mail: {ckavka,proggero,javierma}@unsl.edu.ar

1 Introducción

Esta presentación contiene un resumen de los trabajos más importantes que se están desarrollando actualmente en la Línea de Investigación en Redes Neuronales en el ámbito del LIDIC, el Laboratorio de Investigación en Inteligencia Computacional de la Universidad Nacional de San Luis. Los tres temas principales son control en ambientes cambiantes, control neuro difuso y aplicación de redes neuronales a problemas teóricos. Las secciones siguientes contienen una descripción de cada uno de ellos.

2 Control en ambientes cambiantes

La eficiencia del diseño automático de controladores depende en gran medida de la disponibilidad de un modelo matemático del proceso que se desea controlar. Sin embargo la construcción de modelos apropiados a menudo no es posible debido a la complejidad y no linealidad propia de los procesos. No obstante ello, se han logrado muchos progresos en esta área usando distintos métodos: *redes neuronales*, *sistemas difusos*, *algoritmos evolucionarios*, etc [2].

Las redes neuronales han demostrado ser una alternativa válida para la obtención de controladores, dado que son capaces de aprender cualquier mapeo no lineal. También los algoritmos evolucionarios han recibido mucha atención como métodos para el diseño de controladores, representándolos como redes neuronales en muchos casos [12] [5]. En general, los algoritmos evolucionarios se han utilizado para evolucionar poblaciones de controladores, seleccionando al final de la evolución el controlador definido por el mejor individuo. Este método es adecuado, siempre y cuando el sistema bajo control no cambie en forma dinámica su comportamiento.

Nuestra investigación se ha concentrado en el desarrollo de un algoritmo denominado *Algoritmo evolucionario auto-recomenzable* [1], cuya motivación principal es la evolución de neurocontroladores apropiados para ambientes cuyo comportamiento cambia en el tiempo. El algoritmo almacena poblaciones intermedias de controladores durante la evolución. Cuando disminuye la performance del controlador, el módulo de control de calidad del algoritmo selecciona una de estas poblaciones intermedias para continuar la evolución y obtener un controlador adaptado a las nuevas condiciones ambientales. Este método ha permitido reducir significativamente los tiempos de obtención de controladores basados en redes neuronales RBF en ambientes cambiantes.

3 Control neuro-difuso

El control basado en lógica difusa se utiliza principalmente en problemas de control en áreas donde la complejidad del sistema, el tiempo de desarrollo y el costo son las variables más importantes. El comportamiento de un sistema difuso se define en forma análoga a como lo definiría un operador humano. El controlador consiste en un conjunto de reglas, las cuales constituyen un sistema condicional lingüístico que establece la relación entre entradas y salidas. En muchos casos, los controladores se representan como redes neuronales, con el objetivo de permitir aprendizaje on-line.

La automatización del proceso de construcción de controladores difusos es y ha sido un objetivo muy importante por parte de la comunidad científica. La automatización puede ser parcial, como por ejemplo cuando se actualizan los parámetros de las funciones de pertenencia o la definición de algunas reglas, o total cuando se genera el sistema difuso en forma completa. Muchos investigadores han propuesto estrategias de construcción de sistemas difusos basados en algoritmos evolucionarios. Uno de los primeros trabajos en el área es el de C. Karr [7], que utiliza algoritmos genéticos para la adecuación de las funciones de pertenencia en una base de datos predefinida. Takagi y Lee [13] usan algoritmos genéticos para optimizar las reglas y las funciones de pertenencia al mismo tiempo. Herrera et. al. [4] proponen el método MOGUL, el cual permite construir el controlador difuso combinando un conjunto de reglas definidas por un experto con un conjunto de reglas generadas por un algoritmo evolucionario. Abraham et al. [2] proponen un método general para la construcción de controladores difusos usando redes neuronales y algoritmos evolucionarios, del cual lamentablemente no ofrecen información sobre la clase de operadores genéticos involucrados. Hoffmann [6] propone un método para la actualización fina de las funciones de escala, de las funciones de pertenencia y para la generación de la base de reglas. Clásicos son los trabajos de Kosko et al. [11], en los que propone el modelo SAM, el cual se ha demostrado que es un aproximador universal.

Nuestra investigación en control difuso está vinculada con el aprendizaje on-line de controladores. Es una problemática que adquiere cada vez mayor importancia, dado que la complejidad y la dinámica de los problemas del mundo real requieren sistemas adaptativos inteligentes para su control. El aprendizaje on-line es el aprendizaje de datos cuando el sistema está operando, usualmente en tiempo real, y con datos que pueden estar disponibles solamente por poco tiempo. Existen muchos modelos que han sido definidos para tratar este problema, principalmente vinculados con redes neuronales, dado que el aprendizaje constituye una característica inherente de su definición. Algunos ejemplos de estos modelos son ANFIS, NETFAN y EFUNN [10] entre otros. Sin embargo, existe un problema que no puede ser dejado de lado y está relacionado con el costo computacional de los algoritmos de aprendizaje.

Las tareas de control se realizan en general a través de pequeños *embedded systems*, mientras que la mayoría de los desarrollos de investigación para aprendizaje on-line se evalúan sobre plataformas con alto poder computacional como las computadoras personales. La tendencia no parece cambiar, sino que incluso se refuerza tal como lo expresan las conclusiones del proyecto “Disappearing Computers” de la Unión Europea. Estos pequeños sistemas poseen procesadores dedicados con conexión a red, y una capacidad computacional que no es adecuada para la ejecución de los algoritmos de aprendizaje on-line tradicionales.

Nuestros esfuerzos de investigación están dirigidos a la definición de algoritmos de aprendizaje on-line aplicables a ambientes de microcontroladores. Para ello estamos considerando la aplicación de algoritmos basados en propiedades geométricas de los sistemas difusos, siguiendo la línea de trabajos previos en el tema [9]. A fin de poder evaluar adecuadamente los desarrollos, los experimentos están siendo llevados a cabo en un sistema con un microcontrolador de última generación, con dispositivos conectados a través de un bus estándar en el ambiente industrial.

4 Problemas Teóricos

Por otro lado dentro de la línea se ha trabajado con problemas de origen teórico, en este caso el *problema de satisfactibilidad*. La motivación se debe al hecho que trabajar con instancias de este problema realmente permite entender cuál es la complejidad del mismo. Se ha definido una representación efectiva del problema basada en redes neuronales.

El énfasis está dado en que si es posible hallar un algoritmo que permita encontrar soluciones adecuadas para este problema en tiempos razonables, esto permitiría que otros problemas que no tienen representaciones efectivas puedan entonces ser resueltos, transformándolos en el problema canónico.

El algoritmo DPLL [3] genera un árbol de decisión, que obviamente crece a medida que crece el problema, y en función de las decisiones que debe tomar. En el trabajo que se ha realizado el algoritmo tiene la capacidad de analizar pocos nodos, es decir que necesita tomar pocas decisiones y en general no toma decisiones erróneas [8].

Referencias

- [1] Javier Apolloni, Patricia Roggero and Carlos Kavka. *Evolution of Neurocontrollers in Changing Environments*. Congreso Argentino de Ciencias de la Computación (2002), Buenos Aires, Argentina.
- [2] Abraham A., Jain L. and Kacprzyk J. (Eds.). *Intelligent Systems: Architectures and Perspectives, Recent Advances in Intelligent Paradigms and Applications*. Studies in Fuzziness and Soft Computing, Springer Verlag Germany, (2002) pp. 1-35.
- [3] Stephen A. Cook and David G. Mitchell. *Finding Hard Instances of the Satisfiability Problem: a Survey*. DIMACS (1997).
- [4] O. Cerdón, M.J. del Jesus, F. Herrera, M. Lozano. *MOGUL: A Methodology to Obtain Genetic Fuzzy Rule-based Systems under the Iterative Rule Learning Approach*. Technical Report

#DECSAI-98101. Department of Computer Science and Artificial Intelligence, Universidad de Granada, Spain (1998).

- [5] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics : The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press (2000).
- [6] F. Hoffmann. *Evolutionary Algorithms for Fuzzy Control System Design*. Proceedings of the IEEE, Special issue on industrial innovations using soft computing (2001), pp. 1318,1333.
- [7] C. Karr. *Fuzzy Control of pH using Genetic Algorithms*. IEEE Transactions on Fuzzy Systems 1 (1993), pp. 46-53.
- [8] Carlos Kavka and Patricia Roggero. *Solving 3SAT Problems using Neural Networks*. IASTED, Marbella España (2001).
- [9] Carlos Kavka and Marc Schoenauer. *Voronoi based Function Approximation*. Genetic and Evolutionary Computation Conference (2003), Chicago, Illinois, USA.
- [10] Nikola Kasabov. *Evolving Fuzzy Neural Networks for Supervised - Unsupervised on-line Knowledge - Based Learning*. IEEE Transactions on Man, Machine and Cybernetics, (2001) Vol 31, No. 6, December 2001, pp. 902-918.
- [11] Bart Kosko. *Fuzzy Systems as Universal Approximators*. IEEE Transactions on Computers (1994), vol. 43, no. 11, pp. 1329,1333.
- [12] David Moriarty, Alan Schultz and John Grefenstette. *Evolutionary Algorithms for Reinforcement Learning*. Journal of Artificial Intelligence Research (1999) Vol 11, pp. 241-276.
- [13] M. Lee and H. Takagi. *Integrating Design Stages of Fuzzy Systems using Genetic Algorithms*. Proc. 2nd IEEE International Conference on Fuzzy Systems, San Francisco (1993), pp. 612-617.