

Entrenamiento de Redes Neuronales

Marcelo Alfonso, Javier Apolloni, Carlos Kavka,
Andrea Pérsico, María Soledad Zanglá

Proyecto: Sistemas Inteligentes para Scheduling y Control
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700, San Luis, Argentina

1. Introducción

En la línea Redes Neuronales, dirigida por el Lic. Carlos Kavka, del Proyecto Sistemas Inteligentes para Scheduling y Control, dirigido por el Prof. Raúl Gallard, se está trabajando en temas vinculados con el desarrollo de nuevos algoritmos de entrenamiento de redes neuronales adecuados para su aplicación en distintos contextos.

En particular se consideran dos sublíneas de trabajo:

Evolución de redes neuronales para la obtención de arquitecturas adecuadas para problemas difíciles.

Paralelización de redes neuronales.

2. Evolución de redes neuronales

Los algoritmos evolucionarios son meta-heurísticas de optimización que han ganado una gran reputación en los últimos años debido a sus éxitos en distintas áreas de aplicación [10]. Los algoritmos evolucionarios constituyen, en pocas palabras, una simulación muy rudimentaria del proceso Darwiniano de selección del individuo más adecuado. Los algoritmos evolucionarios han resuelto con éxito problemas difíciles de optimización, desde problemas de optimización paramétricos no continuos hasta instancias muy difíciles de problema de optimización combinatorial, tanto académicos como del mundo real.

La gran flexibilidad de los algoritmos evolucionarios les permite manejar adecuadamente representaciones de espacios de búsqueda no comunes, incluso espacios con información espacial desconocida, como los que se presentan en el campo de la mecánica estructural [11][12].

Los mecanismos de aprendizaje de redes neuronales trabajan en general realizando una búsqueda de los pesos adecuados para una estructura dada, es decir, el número de unidades y el patrón de conexiones en la red son fijos [9]. Existen métodos que permiten la búsqueda de los pesos y la selección de la estructura, pero en general son métodos que trabajan entrenando redes de arquitectura fija, y luego realizando agregado de nuevas unidades o eliminación de unidades superfluas.

El proceso de realizar la búsqueda de una estructura adecuada y además el conjunto de pesos al mismo tiempo, se plantea como un problema extremadamente más difícil dado que el espacio de búsqueda es no estándar. Los algoritmos evolucionarios se plantean entonces como una muy buena posibilidad de

solución en este contexto, dadas sus características.

La propuesta sobre la que se está trabajando consiste en evolucionar poblaciones de componentes de redes neuronales, que son combinadas para su evaluación. De esta forma, ninguna red por sí sola es considerada una solución completa, sino que depende de otras redes. De esta forma, se estimula la especialización de redes en áreas particulares del problema, y al mismo tiempo, se fuerza un sentido de cooperación entre las redes, dado que cada red necesita de otras para poder lograr una solución completa al problema. Este enfoque, propuesto por Miikulainen y Moriarty [1][2][3], y conocido como aproximación simbiótica, ha sido ya utilizado con éxito en trabajos anteriores en esta línea de investigación para la evolución de controladores difusos [14][15].

Se está siguiendo además una aproximación similar a la utilizada en otros dominios de aplicación de algoritmos evolucionarios conocida como la aproximación Michigan [7][10], que se aplica con éxito en Machine Learning, donde los individuos son las reglas y la solución es una base de reglas, y también en el problema inverso de IFS (Iterated Function Systems) en el dominio de procesamiento de imágenes, donde los individuos elementales son mapeos constructivos, y la solución global consiste en un conjunto de esos mapeos. En estas aproximaciones, el mayor problema consiste en el cálculo de la recompensa (fitness) que corresponde a cada individuo.

La gran ventaja de esta aproximación es que en lugar de trabajar con una red de arquitectura fija, se están evolucionando redes de arquitecturas distintas al mismo tiempo, lo que permite encontrar no sólo el conjunto de pesos adecuado, sino además, una arquitectura adecuada para la red.

Los problemas sobre los que se está aplicando esta aproximación son problemas de ajuste de datos (data fitting), donde intencionalmente se proveen áreas del espacio que están muestreadas en forma pobre. También se están analizando problemas de control de robots móviles que deben realizar una serie de tareas en secuencia [5][6][8]. En este caso se plantea la posibilidad de realizar el aprendizaje en forma incremental, partiendo de un problema de una única tarea y agregando gradualmente nuevas tareas.

3. Paralelización de redes neuronales

Una de las desventajas de las redes neuronales consiste en la gran cantidad de tiempo que se necesita para el entrenamiento. Una forma directa de lograr reducir este tiempo, consiste en la paralelización de los algoritmos de aprendizaje [16]. Sin embargo, no siempre los algoritmos se pueden paralelizar en forma simple, y además, la cantidad de comunicación entre las CPUs, hace que la mayoría de las versiones paralelas de estos algoritmos sólo se puedan ejecutar adecuadamente en computadoras paralelas.

En la línea de investigación, se está trabajando en la paralelización de algoritmos de redes neuronales, en particular, a través de un algoritmo que limita la cantidad de información que transita entre los distintas CPUs. De esta forma, los procesadores utilizan a veces información que está desactualizada, pero nunca más allá de un porcentaje, que se define como parámetro del algoritmo.

De esta forma, se trata de permitir la aplicación en clusters de computadoras de algoritmos estándar de paralelización para computadoras paralelas. Uno de los problemas elegidos para probar esta aproximación es el reconocimiento de imágenes. También se considera evaluar la posibilidad de implementar redes modulares. Se están desarrollando aplicaciones usando MPI [17] para su ejecución en una computadora paralela de 32 CPUs y en una red de computadoras Linux.

4. Referencias

- [1] D. E. Moriarty and R. Miikulainen (1996). *Efficient Reinforcement Learning*

- through Symbiotic Evolution*. Machine Learning, 22:11-32.
- [2] D. E. Moriarty and R. Miikulainen (1998). *Hierarchical Evolution of Neural Networks*. Proceedings of the IEEE Conference on Evolutionary Computation.
 - [3] D. E. Moriarty and R. Miikulainen (1998). *Forming Neural Networks through Efficient and Adaptive Coevolution*. Evolutionary Computation 5(4).
 - [4] Amanda J. C. Sharkey (Ed.) (1999). *Combining Artificial Neural Nets*. Springer Verlag
 - [5] T. Kondo, A. Ishiguro, S. Tokura, Y. Uchikawa and P. Eggenberger (1999). *Realization of Robust Controllers in Evolutionary Robotics: A Dynamically Rearranging Neural Network Approach*. CEC Congress on Evolutionary Computation, 1:366-373.
 - [6] O. Chocron and P. Bidaud (1999). *Evolving Walking Robots for Global Task based Design*. CEC Congress on Evolutionary Computation, 1:405-412.
 - [7] Z. Michalewicz (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 3rd ed.
 - [8] S. Nolfi, D. Floreano, O. Miglino and F. Mondada (1994). *How to evolve autonomous robots: Different approaches in evolutionary robotics*. Artificial Life IV. Cambridge, MA.
 - [9] E. Fiesler and R. Beale (Eds.) (1995). *Handbook of Neural Computation*. Oxford University Press and Institute of Physics Publishers.
 - [10] T. Back, D. Fogel and Z. Michalewicz (Eds.) (1997). *Handbook on Evolutionary Computation*. Oxford University Press and Institute of Physics Publishers.
 - [11] M. Schoenauer (1995). *Representations for Evolutionary Optimization and Identification in Structural Mechanics*. In J. Périaux and G. Winter, editors, Genetic Algorithms in Engineer and Computer Science, pages 443-464. John Wiley.
 - [12] M. Schoenauer (1997). *Identification of Mechanical Inclusion*. In D. Dasgupta and Z. Michalewicz, editors, Evolutionary Computation in Engineering, pages 477-494. Springer verlag.
 - [13] P. Collet, E. Lutton, F. Raynal and M. Schoenauer (1999). *Individual GP: An Alternative Viewpoint for the Resolution of Complex Problems*. Proceedings of the First Genetic and Evolutionary Computation Conference, MIT Press.
 - [14] Carlos Kavka, María Liz Crespo, Marcelo Cena, Wu Geng feng and Fu Zhong quian (1998). *Fuzzy systems generation through symbiotic evolution*. Proceedings of the International Symposium on Engineering of Intelligent Systems EIS'98, Tenerife, España.
 - [15] Carlos Kavka, María Liz Crespo, Marcelo Cena, Wu Geng feng and Fu Zhong quian (1999). *A fuzzy controllers development tool based on evolutionary techniques*. Proceedings of CEC Congress on Evolutionary Computation.
 - [16] Gropp William, Marc Snir, William Gropp, Bill Nitzberg and Ewing Lusk (1998). *Mpi: The Complete Reference: The Mpi Core/the Mpi Extensions (Scientific and Engineering Computation Series)*. Mit Press.
 - [17] Barry Wilkinson and C. Michael Allen (1998). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. Prentice-Hall.