



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

---

# Sistema evolutivo híbrido para la construcción de Redes de Neuronas

---

TRABAJO FIN DE MÁSTER  
MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL

AUTOR: Carlos Vázquez Losada  
TUTOR: Daniel Manrique Gamo

20 de julio de 2019



# Agradecimientos

Deseo expresar mi agradecimiento, en primer lugar, a Daniel, un investigador incansable y un profesor excepcional, pionero en Computación Evolutiva en España y protagonista indudable de la evolución de la Programación Genética. Gracias por dedicarme tu tiempo y adaptarte a mis horarios, y gracias por despertar en mí el interés en la Computación Evolutiva con tus clases y con tu implicación.

A mis padres, Isabel Losada y Fco. Javier Vázquez, por su amor, comprensión y por su apoyo incondicional en todo aquello que me he propuesto.

A Cristina, espectadora de mis éxitos y mi acompañante. Siempre sabes que decir y cómo complementarme. Te quiero.



# Resumen

Este Trabajo de Fin de Máster consiste en estudiar la idoneidad de la Computación Evolutiva para la generación de Redes de Neuronas Artificiales. En concreto, en un estudio de la Programación Genética Guiada por Gramáticas y su utilización para la creación de la mejor Red de Neuronas Artificiales obtenida de un problema determinado, realizando para ello experimentos de entrenamiento total y parcial de cada una de las redes candidatas.

La Programación Genética es una técnica evolutiva (y por tanto, inspirada en la biología) y que es utilizada en problemas de optimización cuyas soluciones son programas informáticos. La Programación Genética Guiada por Gramáticas extiende las posibilidades de la Programación Genética tradicional con la introducción de las gramáticas, que permiten crear individuos sintácticamente válidos.

Estas gramáticas permiten la confección de arquitecturas de Redes de Neuronas Artificiales que son válidas, dada cualquier número de neuronas en la capa de entrada y en la capa de salida. El resultado de una ejecución nos devolverá la arquitectura de red que mejor se amolde al problema dado, teniendo en cuenta las propias peculiaridades de la gramática utilizada. Además, se estudia si el resultado es el mismo entrenando individuos de forma total o parcial.



# Summary

This Master Thesis Dissertation consists in a research about how appropriate is to use Evolutionary Computation for creating Artificial Neural Networks. A technique of Evolutionary Computation is used for this purpose; Grammar-Guided Genetic Programming. It will be used for looking the best Artificial Neural Network for a concrete dataset, by checking both partially and fully trained networks.

Genetic Programming is an evolutionary technique (it is inspired by biology as Evolutionary Computation does) and it is used for solving optimization problems whose solution is a computer program. The Grammar-Guided Genetic Programming extends Genetic Programming by adding grammars, that allow to create valid individuals.

These grammars allow creating Neural Network architectures that are all valid, given any number of neurons in both input and output layers. The result of an execution would be the architecture that best fixes (in terms of accuracy) the given data.





# 1. Introducción

La optimización matemática estudia un tipo concreto de problemas donde se desea elegir el mejor entre un conjunto de elementos. El problema clásico (Boyd and Vandenberghe, 2004) consiste en maximizar o minimizar una función objetivo que representa o mide la calidad de las decisiones. Además, esta función está sujeta a un conjunto de restricciones que acotan el espacio de soluciones e intenta simplificar su obtención.

Para resolver estos problemas, contamos con algoritmos de optimización, métodos iterativos y heurísticas. Los algoritmos son técnicas que se ejecutan durante un número determinado de pasos y que devuelven el mejor valor hasta entonces. El primero de ellos fue el algoritmo de Simplex (Dantzig, 1990). Los métodos iterativos buscan la convergencia hacia una solución determinada. El primer método fue el Método de Newton (Nocedal et al., 1999). Las heurísticas (Polya, 1945), en cambio, aproximan la solución en el caso de que previamente no haya habido convergencia.

La Computación Evolutiva es una familia de heurísticas disponibles para la resolución de problemas de optimización. En concreto, la Programación Genética (Koza, 1992) surge por la necesidad de extender la optimización para involucrar programas informáticos. La Computación Evolutiva así como todas sus técnicas surgen inspiradas en la propia evolución biológica de los seres vivos.

Por otro lado, las Redes de Neuronas Artificiales (McCulloch and Pitts, 1943) son técnicas de Aprendizaje Automático (Samuel, 1959) inspiradas en las redes de neuronas propias de los seres vivos. El objetivo de este modelo es la resolución de problemas tal y como lo hace el cerebro humano. Son especialmente utilizadas en los ámbitos de Visión Artificial (Roberts, 1965) y en el Reconocimiento de Voz (Waibel et al., 1989), aunque también son ampliamente usadas para problemas de Aprendizaje Supervisado básicos como los conformados por información proveniente de ficheros, y es en este tipo de clasificación en la que se centra este trabajo.

Estas dos herramientas, ambas inspiradas en la biología existente y observable por el ser humano, no son opuestas sino que se complementan. La Computación Evolutiva permite resolver problemas de optimización, mientras que las Redes de Neuronas Artificiales son indicadas para el Aprendizaje Automático. Disponer de una heurística de optimización que en cuestión de minutos sea capaz de determinar la Red de Neuronas que mejor resuelve un problema supone un gran avance en su utilización en problemas de Inteligencia Artificial, ya que, hasta ahora, la selección de la arquitectura es difusa y se basa en la experiencia previa, con conclusiones muchas veces no demasiado certeras.

El objetivo principal de este trabajo es estudiar la idoneidad de la Computación Evolutiva para la generación de Redes de Neuronas Artificiales. Para ello, utiliza-

remos una de las técnicas de esta heurística, la Programación Genética Guiada por Gramáticas, para tal fin. En concreto, se establecen varios objetivos:

1. Implementar un Programa Genético que lidie con Redes de Neuronas y que permita la utilización de la gramática.
2. Determinar si la Computación Evolutiva resulta útil para la resolución de un problema de optimización todavía no resuelto: cuál es la Red de Neuronas que mejor se ajusta a un problema dado.
3. Comprobar que la gramática que se utiliza en este trabajo satisface este fin.
4. Afirmar si, en el procedimiento seguido, entrenar las redes candidatas (individuos del programa) de forma parcial (según una métrica establecida), es igual de significativo que entrenar las redes candidatas de forma total (hasta que haya sobreentrenamiento).

Este documento se estructura en siete secciones después de esta introducción. Las dos primeras se centran en la *Computación Evolutiva* y en las *Redes de Neuronas Artificiales*, respectivamente. La tercera sección trata sobre la *construcción de Redes de Neuronas*, seguida del *planteamiento del problema*, en la que se describe con profundidad el propósito de este trabajo y la necesidad real de él. A esta sección le sigue la *solución propuesta*, que contiene la metodología y el procedimiento seguido, exponiendo los *resultados* obtenidos en la séptima sección. Finalmente, el trabajo finaliza con las *conclusiones y líneas futuras* propuestas.

## 2. Computación Evolutiva

La Computación Evolutiva es un compendio de heurísticas destinadas a la resolución de problemas de optimización y que todas ellas están inspiradas en la propia evolución biológica de los seres vivos. En este capítulo se ofrecen, en primer lugar, los *inicios y el estado del arte* de la Computación Evolutiva. Seguidamente, una *definición* y que finaliza hablando sobre la *Programación Genética*.

### 2.1. Inicios y estado del arte

La Computación Evolutiva tiene su primera aparición en la literatura científica gracias a los trabajos de Box (Box, 1957), Friedberg (Friedberg, 1958), (Friedberg et al., 1959) y Bremermann (Bremermann, 1962). Sin embargo, no se consiguieron grandes avances debido a la pobre metodología todavía sin desarrollar y a las limitaciones computacionales de la época.

Algunos años después surgen los primeros desarrollos metodológicos en una década de destacable logro científico. El trabajo de Fogel (Fogel et al., 1966) sienta las bases de la Programación Evolutiva (*evolutionary programming* en inglés) y el de Holland (Holland, 1967) las de los Algoritmos Genéticos (*genetic algorithms* en inglés). También, en esa misma época, las Estrategias Evolutivas (*evolution strategies* en inglés) fueron introducidas por Rechenberg (Rechenberg, 1965) y Schwefel (Schwefel, 1965).

Una década después de los primeros descubrimientos, alrededor de los años 80, los avances computacionales permitieron aplicar las técnicas evolutivas descubiertas para resolver problemas de optimización del mundo real. En esa misma década, los estudios de Cramer (Cramer, 1985) y Koza (Koza, 1988) desembocaron en la aparición de una nueva técnica perteneciente a la Computación Evolutiva: la Programación Genética (*genetic programming* en inglés). También se sucedieron una gran cantidad de conferencias internacionales y talleres centrados en aspectos teóricos de los Algoritmos Genéticos (Grefenstette, 1985), (Grefenstette, 1987) o (Schaffer, 1989), entre muchos otros. Estos años destacaron también por la aparición de otras técnicas de Computación Evolutiva como la Vida Artificial (*artificial life* en inglés) abreviada muy comúnmente como *A-Life* (Langton, 1986), los Sistemas Inmunitarios Artificiales (*artificial immune systems* en inglés) (Farmer et al., 1986), la Inteligencia de Enjambre (*swarm intelligence* en inglés) (Beni and Wang, 1989) y los Algoritmos Meméticos (Moscato, 1989).

Para 1990, era innegable admitir que parte de la comunidad científica del mundo ponía sus ojos en la Computación Evolutiva. A las conferencias anteriormente ex-

puestas le siguieron las cuatro conferencias de la IEEE sobre Computación Evolutiva, que asentaron esta rama de estudio como una de las articulaciones de la Inteligencia Artificial y herramienta indispensable para la resolución de problemas de optimización en el ámbito académico y empresarial. La conferencia de Programación Genética (Koza et al., 1996) tuvo gran éxito y aceptación, al igual que otras sobre desarrollos metodológicos sobre los Algoritmos Genéticos (Rawlins, 1991), (Whitley, 1993) y (Vose, 1995), que se consolidó como la técnica de Computación Evolutiva más destacable. Esta década se cierra con la llegada de nuevas técnicas de Computación Evolutiva: los Algoritmos Culturales (*cultural algorithms* en inglés) (Reynolds, 1994), la Evolución Diferencial (*differential evolution* en inglés) (Storn, 1996) y (Storn and Price, 1997) y la Evolución Gramatical (*grammatical evolution* en inglés) (Ryan et al., 1998).

### 3. Redes de Neuronas Artificiales

## 4. Construcción de Redes de Neuronas

## 5. Planteamiento del problema

## 6. Solución propuesta



## 7. Resultados

## 8. Conclusiones y líneas futuras

# Bibliografía

- Beni, G. and Wang, J. (1989). Swarm intelligence in cellular robotic systems. *Proceedings of the NATO Advance Workshop on Robots and Biological Systems*. 102:703-712.
- Boyd, S. and Vandenberghe, L. (2004). Convex Optimization. *Cambridge University Press 2004*: 129.
- Box, G.E.P. (1957). Evolutionary operation: A method for increasing industrial productivity. *Journal of the Royal Statistical Society. Series C*. 6(2):81-101.
- Box, G.E.P. and Draper, N.P. (1969). *Evolutionary Operation. A method for Increasing Industrial Productivity*.
- Bremermann, H.J. (1962). Optimization through evolution and recombination. *Self-Organizing Systems 1962*: 93-106.
- Cramer, N.L. (1985). A representation for the Adaptive Generation of Simple Sequential Programs. *Proceedings of the First International Conference on Genetic Algorithms and the Applications*: 183-187.
- Dantzig, G.B. (1990). Origins of the simplex method. *A history of scientific computing*, pages 141-151.
- Farmer, J.D., Packard, N.H. and Perelson, A.S. (1986). The Immune System, Adaptation, and Machine Learning. *Physica D: Nonlinear Phenomena*. 22(1): 187-204.
- Fogel, L.J., Owens, A.J. and Walsh, M.J. (1966). *Artificial Intelligence through Simulated Evolution*.
- Fraser, A.S. (1957). Simulation of Genetic Systems by Automatic Digital Computers. *Australian journal of biological sciences*. 10:484-499.
- Friedberg, R.M. (1958). A learning machine: part I. *IBM Journal of Research and Development*. 2(1):2-13.
- Friedberg, R.M., Dunham, B. and North, J.H. (1959). A learning machine: part II. *IBM Journal of Research and Development*. 3(3):282-287.
- Grefenstette, J.J. (1985). *Proceedings of the First International Conference on Genetic Algorithms and the Applications* (Hillsdale, NJ: Lawrence Erlbaum).

- Grefenstette, J.J. (1987). *Proceedings of the Second International Conference on Genetic Algorithms and the Applications* (Hillsdale, NJ: Lawrence Erlbaum).
- Holland, J.H. (1962). Nonlinear environments permitting efficient adaptation. *Computer and Information Sciences II*: 147-164.
- Koza, J.R. (1988). Non-Linear Genetic Algorithms for Solving Problems. *University States Patent 4935877*.
- Koza, J.R. (1992). *Genetic Programming 1992* (Cambridge, MA: MIT Press).
- Koza, J.R., Goldberg, D.E., Fogel, D.B. and Riolo, R.L. (1996). *Genetic Programming 1996* (Cambridge, MA: MIT Press).
- Langton, C.G. (1986). Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*. 22(1):120-149.
- McCulloch, W.S. and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. 5(4):115-133.
- Moscato, P. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms. *Caltech Concurrent Computation Program*: 158-179.
- Nocedal, J. and Wright, S.J. (1999). Numerical optimization. *Springer-Verlag*.
- Polya, G. (1945). How to Solve It. *Princeton University Press 1945*.
- Rawlins, G.J.E. (1991). *Foundations of Genetic Algorithms* (San Mateo, CA: Morgan Kaufmann).
- Rechenberg, I. (1965). Cybernetic Solution Path of an Experimental Problem. *Royal Aircraft Establishment Library Translation 1122*.
- Reynolds, R.G. (1994). An Introduction to Cultural Algorithms. *Proceedings of the 3rd Annual Conference on Evolutionary Programming*: 131-139.
- Roberts, L. (1965). Machine perception of 3D solids. *Optical and Electro-Optical Information Processing*. 9:159-197.
- Ryan, C., Collins, J.J. and O'Neil, M. (1998). Grammatical Evolution: Evolving Programs for an Arbitrary Language. *Proceedings of the First European Workshop on Genetic Programming*: 83-96.
- Samuel, A.L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*. 3(3):210-229.
- Schaffer, J.D. (1989). *Proceedings of the Third International Conference on Genetic Algorithms and the Applications* (San Mateo, CA: Morgan Kaufmann).
- Schwefel, H-P. (1965). Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik.

- Storn, R. (1996). On the usage of differential evolution for function optimization. *Biennial Conference of the North American Fuzzy Information Processing Society*: 519-523.
- Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 11(4):341-359.
- Vose, M.D. and Whitley, L.D. (1995). *Foundations of Genetic Algorithms 3* (San Francisco, CA: Morgan Kaufmann).
- Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K.J. (1989). Phone-me recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 37(3):328-339.
- Whitley, L.D. (1993). *Foundations of Genetic Algorithms 2* (San Mateo, CA: Morgan Kaufmann).