

Guía Práctica No. 6: Algoritmos Genéticos y Redes Neuronales

Esta guía práctica corresponde a Algoritmos Genéticos y Redes Neuronales.

Para resolver los problemas correspondientes a algoritmos genéticos se utiliza el framework JGAP <https://sourceforge.net/projects/jgap/>.

Para la configuración y entrenamiento de redes neuronales se utiliza la librería *Scikit-Learn* <https://scikit-learn.org/> y puede acceder a la implementación de un modelo a través del classroom de github.

Parte de esta práctica cuenta con esquemas de programación y soluciones implementadas.

Esta práctica no tiene entrega formal. Su resolución es opcional.

1. Considere el problema de dar cambio por C centavos, utilizando monedas de valores $d_1 > d_2 > \dots > d_k$, y sabiendo que se cuenta con M_i monedas de valor d_i . Diseñe un algoritmo utilizando algoritmos genéticos, y el framework JGAP, que resuelva el problema de dar cambio. Su programa debe dar como resultado cuáles son las monedas que componen el cambio, y debe respetar la restricción de que no pueden darse en el cambio más de M_i monedas de valor d_i . Puede aprovechar el ejemplo del tutorial de JGAP para guiar su solución.
2. Considere el conocido *Problema de la Mochila*, diseñe un algoritmo para resolver este problema usando algoritmos genéticos, e implemente el mismo a través del framework JGAP. Su programa debe maximizar el valor total de ítems seleccionados, sin exceder la capacidad máxima de la mochila.
3. Considere el *Problema de las 8 Reinas*, que consiste en ubicar a 8 reinas en un tablero de ajedrez de 8×8 , de manera que las mismas no se ataquen entre sí. Diseñe un algoritmo para resolver este problema usando algoritmos genéticos, e implemente el mismo a través del framework JGAP. Su programa debe dar como resultado la mejor ubicación encontrada para las reinas, y el número de reinas que se atacan entre sí en dicha configuración.
4. Utilizando la librería *Scikit-Learn* y la estructura de la red neuronal que implementa el operador *OR* definido en el repositorio de la materia (exercise-set-6/or_model.py), realice pruebas modificando los algoritmos de entrenamiento y los valores de los parámetros de las distintas funciones utilizadas para entrenar la red (revise la documentación de la librería).
5. Utilizando la librería *Scikit-Learn* y teniendo como guía la implementación del operador OR:
 - modele y entrene una red neuronal que aprenda la función *AND*
 - modele y entrene una red neuronal que aprenda la función *XOR*
6. Utilizando la librería *scikit-learn*, modele una red sobre el clima, que permita predecir la temperatura promedio máxima en un año determinado, entrenando la red con un porcentaje del 60% de los datos de entrada y luego medir precisión de la red con el 40% restante. Puede encontrar una base datos con el clima histórico de la ciudad de Córdoba en el repositorio de la materia.