

Taller No °4 Programación genética

1. EJERCICIOS

1.1. Descargue MEPX, <https://www.mepx.org/>, estúdielo y corra uno de los ejemplos que trae.

Se ejecuta y soluciona el problema `last_2_digits_of_2_to_power_n.xml` el cual es un desafío relacionado con series temporales y cálculos numéricos en MEPX. El nombre del archivo sugiere que el objetivo es predecir los últimos dos dígitos de 2 elevado a la potencia de n , donde " n " probablemente representa un número en una secuencia o serie.

El objetivo principal de este problema en MEPX puede ser generar un modelo matemático que prediga los últimos dos dígitos de la secuencia resultante de 2 elevado a la potencia de " n ".^{en} función de los valores previos de " n ". Por ejemplo, para $n=1$, $2^1 = 2$; para $n=2$, $2^2 = 4$; para $n=3$, $2^3 = 8$, y así sucesivamente.

Para ello se vale de una lista de entrenamiento como se muestra a continuación:











Training data	Validation data	Test data	Advanced commands	Problem description
 Load training data		#	Target0	
 Delete training data		1	1	
 Export training data		2	2	
 Find values		3	4	
 Replace values		4	8	
 Shuffle		5	16	
 Move to test		6	32	
 Move to validation		7	64	
 Paste from Clipboard		8	28	
 Num. outputs		9	56	
		10	12	

Figura 1: Lista de entrenamiento.

Los procesos utilizados u operadores se muestran en la siguiente imagen al igual que el tipo de proceso empleado para la ejecución.

The configuration window is divided into several panels:

- Data type:** Integer numbers (selected).
- Problem type:** Time-series (selected).
- Error measure:** Mean Absolute Error (selected).
- Time series:** Window size: 1, Mode: Predict new data (selected), Num. predictions: 10.
- Functions:** Addition, Subtraction, Multiplication, Division, Power, Sqrt, Exp, Pow10, Ln, Log10, Log2, Floor, Ceil, Abs, Inv (1/x), Neg (-x), X^2, Min, Max, Sin, Cos, Tan, ASin, ACos, ATan, If a<0?b:c, If a<b?c:d, If a<0 or b<0 ? c:d, If a<0 xor b<0 ? c:d, Modulus.
- Parameters:** Num. subpopulations: 2, Subpopulation size: 100, Code length: 50, Crossover probability [0..1]: 0,9, Crossover type: Uniform (selected), Mutation probability [0..1]: 0,01, Tournament size: 2, Probabilities: Functions: 0,5, Variables: 0,5, Constants: 0, Num. generations: 1000.
- Constants:** Type: Automatically generated (selected). User defined constants: (empty list). Automatically generated constants: Number of constants: 5, Min initial interval: 0, Max initial interval: 1, May evolve: (checked), Evolve outside initial interval: (unchecked), Max delta: 1.
- Runs:** Random seed: 0, Num. runs: 10, Num. threads: 2, Training subset: Random subset size (%): 100, Num. generations for a subset: 1, Use validation data: (checked).

Figura 2: Configuración del modelo.

Una vez ejecutado el programa se tiene una ventana de resultados donde se muestra el valor obtenido y se compara con el esperado mostrando el error en cada iteración.

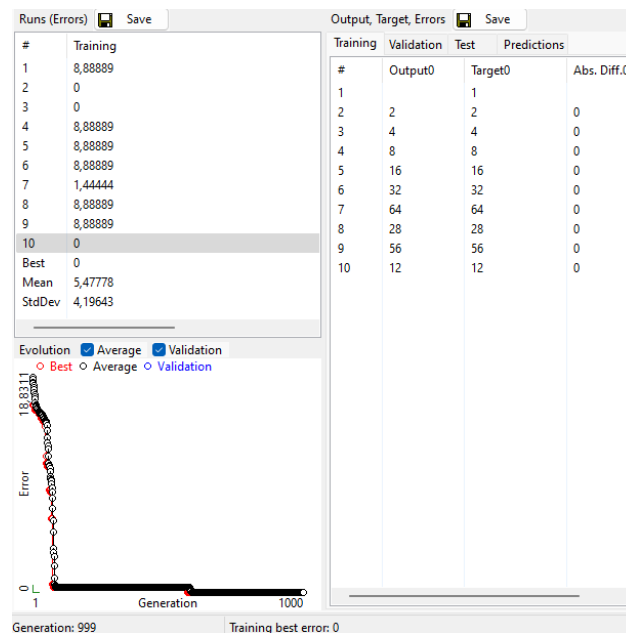


Figura 3: Tabla de resultados.

Finalmente, se tiene un gráfico que compara el valor esperado con el obtenido y se evidencia que salvo el primer dato el cual no arrojó ningún valor, los demás coinciden con lo previsto.

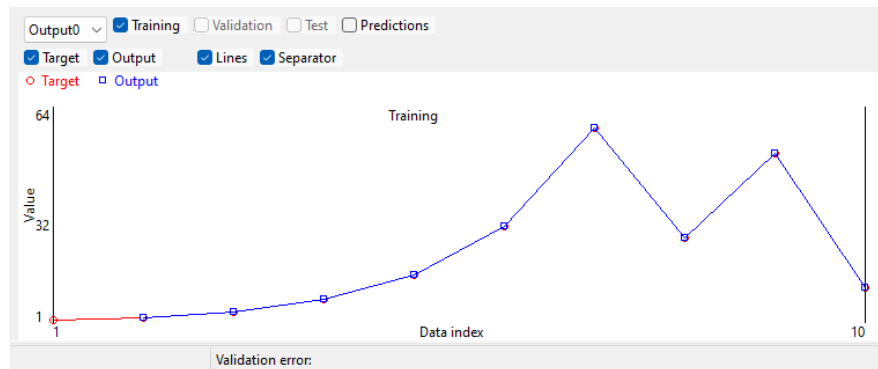


Figura 4: Comparativa gráfica de los resultados y lo esperado.

1.2. Suponga que el valor de la TRM depende del valor de acción de Ecopetrol, del valor de la acción del Banco de Colombia y del precio del barril de petróleo. Utilice MEPX para encontrar una fórmula que calcule la TRM para el día siguiente. Para el aprendizaje utilice 60 datos de cada componente.

El desarrollo de este punto se encuentra en el siguiente enlace:

<https://github.com/YeisonVR/Trabajos-IA-y-MR/tree/main>

Para este punto se genera una lista de los valores históricos de acciones de Ecopetrol, Bancolombia y del precio del barril de petróleo, del mismo modo se establece una TMR objetivo tomada nuevamente de un periodo de 3 meses al igual que los demás datos y se construye así los datos de entrenamiento, luego se emplea el tipo de problema de regresión para generar un modelo predictivo de TMR obteniendo así la siguiente gráfica.

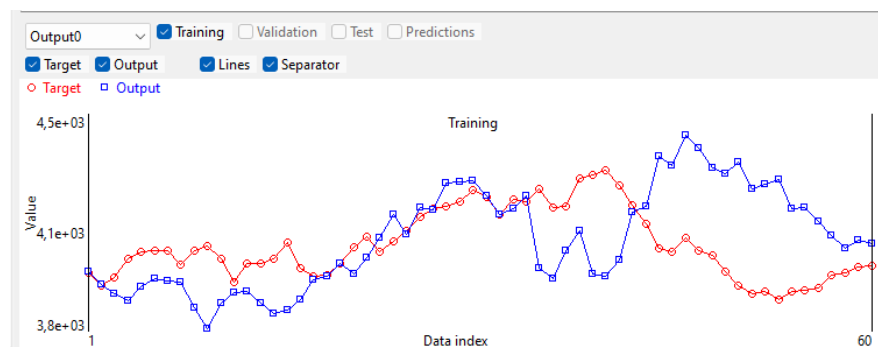


Figura 5: Comparativa gráfica de los resultados y lo esperado.

Los datos en excel así como el modelo en formato .xml se encuentran en la carpeta Taller 4 del github mencionado anteriormente.

- 1.3. Sponga que desea utilizar Programación Genética para encontrar el diseño de un circuito lógico, tome como, ejemplo el codificador de 7 segmentos. Describa el conjunto de terminales, el conjunto de funciones y la función de aptitud. Use una librería de Python.**

El desarrollo de este punto se encuentra en el siguiente enlace:
<https://github.com/YeisonVR/Trabajos-IA-y-MR/tree/main>

- 1.4. Sponga que tiene un robot que le entrega galletas al grupo de ingenieros de diseño de robots. Programe por PG el recorrido del robot, teniendo en cuenta que cada vez que un ingeniero recibe una galleta gana puntos. Los ingenieros están distribuidos en una sala cuadrada. Defina, conjunto de terminales, conjunto de funciones y función de aptitud.**

El desarrollo de este punto se encuentra en el siguiente enlace:
<https://github.com/YeisonVR/Trabajos-IA-y-MR/tree/main>

- 1.5. Lea el artículo referenciado en [11], haga un resumen de un par de hojas y obtenga unas conclusiones del uso de PG en la programación de máquinas.**

El presente artículo constituye una revisión exhaustiva acerca del uso de la programación genética (GP) en la planificación de la producción. Su principal objetivo radica en presentar un esquema unificado para la automatización del diseño de estrategias heurísticas en la planificación de la producción mediante el empleo de la GP. Este esquema se desglosa en cuatro componentes principales: representación, operadores genéticos, mecanismo de búsqueda y función de aptitud. Los autores proporcionan un análisis detallado de las diversas alternativas de diseño para cada componente y presentan ejemplos provenientes de configuraciones variadas empleadas en estudios previos.

En su inicio, el documento ofrece una breve introducción a la programación de producción y a la programación genética, dirigida a aquellos investigadores que incursionan por primera vez en estas áreas. Posteriormente, los autores se sumergen en un análisis exhaustivo de los cuatro componentes del esquema de la GP. Discuten distintos tipos de representaciones, incluyendo aquellas basadas en árboles y grafos, evaluando las ventajas y desventajas inherentes a cada una. Asimismo, examinan varios operadores genéticos, como la mutación y el cruce, y exploran sus implicaciones en el proceso de búsqueda.

Los autores detallan también diversos mecanismos de búsqueda, como la escalada de colinas y el recocido simulado, contextualizando su idoneidad frente a distintos tipos de problemas. De manera complementaria, abordan las funciones de aptitud, cuya finalidad radica en evaluar la calidad de las soluciones candidatas, y presentan ejemplos ilustrativos de diferentes tipos de funciones de aptitud aplicadas en investigaciones anteriores.

El documento integra además un análisis exhaustivo de estudios previos que han utilizado la GP en la planificación de producción. Los autores sintetizan los principales hallazgos de dichos estudios y ofrecen perspectivas acerca de la efectividad de distintas configuraciones del esquema de GP. A su vez, reflexionan sobre las limitaciones de dichos estudios y proponen líneas para futuras investigaciones.

En resumen, esta obra constituye una fuente invaluable para investigadores interesados en emplear la GP en la planificación de la producción. El esquema unificado presentado en esta revisión puede asistir a los investigadores en la comprensión de los componentes clave y las configuraciones pertinentes para esta labor. El repaso de investigaciones previas brinda información relevante sobre la efectividad de distintos enfoques y puede orientar futuras investigaciones en este campo en evolución.

La programación genética (GP) se destaca como una herramienta potente para el diseño automatizado de estrategias heurísticas en la planificación de la producción. Múltiples estudios han evidenciado que las estrategias evolucionadas mediante GP superan a numerosas heurísticas existentes concebidas manualmente en la literatura. La flexibilidad inherente a la GP posibilita la creación de estrategias altamente sofisticadas para abordar entornos de producción complejos y dinámicos.

Este artículo introduce un marco unificado para el diseño automatizado de estrategias heurísticas en la planificación de la producción a través de la GP. El propósito del marco es proporcionar una visión global a los investigadores sobre la aplicabilidad de la GP en esta labor y sus componentes clave. Además, el marco se emplea para facilitar el análisis y debate de estudios existentes en este campo.

La revisión de investigaciones previas brinda información valiosa sobre la eficacia de diferentes configuraciones del marco de GP. Los autores reflexionan sobre las limitaciones de las investigaciones previas y plantean direcciones para futuras investigaciones.

En conclusión, la programación genética es una técnica prometedora para el diseño automatizado de heurísticas de programación de producción. El marco unificado presentado en este documento puede ayudar a los investigadores a comprender los componentes clave y las configuraciones necesarias para esta tarea. La revisión de estudios anteriores proporciona información valiosa sobre la efectividad de diferentes enfoques y puede guiar futuras investigaciones en esta área.

Referencias

- [1] Nguyen, Su, Yi Mei1 y Mengjie Zhang: *Genetic programming for production scheduling: a survey with a unified framework*. SpringerLink, 2017.