

## Proyecto de biónica cognitiva

El proyecto se ha dividido en tres fases:

- Fase 1: Desarrollo de un algoritmo para la generación de matrices anidadas de números primos, para lo cual se han desarrollado distintos programas de construcción de matrices y se han llevado a cabo distintos estudios sobre ellas para intentar identificar propiedades de las matrices que nos permitan construirlas de la manera más eficiente posible.
- Fase 2: Experimentos con distintos algoritmos de Machine Learning con el objetivo de que sean capaces de generar un mecanismo de construcción de este tipo de matrices.
- Fase 3: A vista de la falta de resultados satisfactorios en la fase 3, se ha intentado un nuevo enfoque buscando aplicar la tecnología de memcomputing además de los algoritmos de ML. Para esta fase se han llevado a cabo más estudios de las propiedades de las matrices que nos ayuden a la identificación de las mismas por estos algoritmos.

En la siguiente lista se explican los programas y resultados correspondientes a cada fase. Se han marcado en rojo los que pertenecen simultáneamente a las fases 1 y 3

### Fase 1

Dentro de esta fase encontramos distintos programas para la construcción de matrices y resultados de los distintos estudios realizados sobre estas.

- Programas de generación de matrices:
  - Programas que generan matrices de 3x3:
    - Programas que generan matrices de primos
      - crear\_matrices\_anidadas\_3\_de\_primos\_solo\_una\_matriz\_por\_central.py
        - Este programa genera matrices anidadas de primos pero solo crea una matriz por número central.
      - crear\_matrices\_anidadas\_3\_de\_primos\_solo\_una\_matriz\_por\_central\_parallelizado\_usando\_multiprocessing.py
        - Igual que el programa anterior pero de manera paralelizada para disminuir el tiempo de procesamiento
      - crear\_matrices\_anidadas\_3\_de\_primos\_usando\_numpy\_y\_sympy.py
        - Genera matrices anidadas de primos de 3x3. Crea todas las matrices posibles por cada elemento central. Muy útil para estudiar matrices análogas y generar grandes cantidades de datos de estudio.
    - Programas que generan matrices de primos y compuestos
      - crear\_matrices\_anidadas\_3\_a\_partir\_de\_c\_y\_diferencias\_(multiplos\_de\_6).py
        - Este programa genera matrices partiendo de las diferencias entre los elementos. Nos ayuda a estudiar el comportamiento de las diferencias entre matrices

- crear\_matrices\_anidadas\_3\_de\_8\_primos\_generadas\_por\_no\_multiplos\_de\_2\_3\_o\_5\_usando\_numpy\_y\_sympy.py
    - Este programa genera matrices de 8 primos a partir de un central compuesto. Lo utilizamos para estudiar las matrices en que solo el central es compuesto.
  - crear\_matrices\_anidadas\_3\_usando\_numpy\_y\_sympy.py
    - Este programa crea matrices anidadas de tres sin distinguir entre primos y no primos. Lo utilizamos para estudiar matrices compuestas por distinto n
  - Programas que generan matrices de 5x5:
    - crear\_matrices\_anidadas\_5\_de\_primos.py
      - Este programa genera matrices anidadas de primos de 5x5. Crea todas las matrices posibles por central.
- Estudios y resultados
  - 1. Estudio relación entre elementos
    - Este estudio corresponde a la fase uno porque se centra en el estudio de las relaciones que existen entre los elementos de una matriz anidada de números primos, lo que es clave para la construcción de un algoritmo eficiente de construcción de matrices
  - 2. Estudio de matrices según las diferencias entre los elementos
    - Las diferencias entre los elementos de las matrices (en concreto el hecho de que todas sean múltiplos de 6) es uno de las características claves a la hora de desarrollar un algoritmo eficiente
  - 4. Estudio números que generan matrices
    - Es importante conocer qué números son capaces de generar matrices anidadas de números primos para elaborar un algoritmo que encuentre las matrices generadas por dichos números
  - 5. Estudio descomposición de pares
    - Relacionado con el punto anterior, las posibles descomposiciones de pares de un número son las que determinan si dicho número va a ser capaz de generar matriz o no
  - 6. Estudio del número de matrices análogas
    - Relacionado con el estudio 4 también, no solo es interesante saber qué números generan matrices sino cuántas matrices genera cada uno de ellos para saber qué es lo que esperamos encontrar exactamente con nuestro algoritmo.
  - 9. Estudio de 8 matrices y un “pseudoprimo”
    - Una de las claves del experimento es buscar propiedades que distingan las matrices de primos de aquellas que no lo son, esto es importante cuando desarrollamos un algoritmo que solo busque matrices de primos
  - 10. Estudio de matrices de primos y no primos según número de primos
    - Igual que en el punto anterior, no solo es interesante estudiar las matrices con un número compuesto sino si existen diferencias entre matrices con distinto número de primos
  - 11. Estudio de matrices de primos y no primos según múltiplos
    - Como los primos, precisamente, se caracterizan por no tener divisores, una forma de buscar un algoritmo más eficiente es a través de la

relación entre los distintos múltiplos de los diversos primos que aparecen en las matrices

- 14. Estudio volumen de pares

- Este estudio tiene el objetivo de determinar qué volumen mínimo de pares de primos mínimos necesitamos utilizar en nuestro algoritmo para poder generar matrices de distintos tamaños

- Muestra de matrices generadas

- En la carpeta se incluye también una muestra de matrices generadas por el algoritmo definitivo que va desde el rango 17 hasta el rango 901

## Fase 2

- Algoritmos de procesamiento de lenguaje natural

- Resultados

- Experimentos-gpt

- Se incluyen los resultados de los entrenamientos llevados a cabo por distintos algoritmos de la familia GPT

- Alpha Zero

- Programas modificados

- DeepReinforcementLearning

- Adaptación del algoritmo

<https://github.com/AppliedDataSciencePartners/DeepReinforcementLearning>

- Alpha-zero-general

- Adaptación del algoritmo

<https://github.com/suragnair/alpha-zero-general>

- Resultados

- Experimentos-alphazero:

- Resultados de los entrenamientos llevados a cabo de los algoritmos de Alpha Zero

- SATNet

- Programas modificados

- SATNet

- Adaptación del algoritmo

<https://github.com/locuslab/SATNet>

- Resultados

- Experimentos-SATNet

- Resultados de los entrenamientos llevados a cabo con el algoritmo de SATNet

- Programas de tratamiento de imágenes

- Programa modificado

- Generative\_inpainting

- Adaptación del algoritmo

- [https://github.com/JiahuiYu/generative\\_inpainting](https://github.com/JiahuiYu/generative_inpainting)

- Resultados

- Experimento-generative-inpainting

- Resultados de los entrenamientos llevados a cabo por el algoritmo de generative inpainting

### Fase 3

- 3. Estudio autovalores de las matrices
  - Este estudio tiene el objetivo de analizar las características de los autovalores y los autovectores de las matrices anidadas de primos, lo cual puede ser una propiedad clave para que los modelos de machine learning y memcomputing identifiquen la matriz
- 7. Estudio de las frecuencias de aparición de valores en las matrices de primos
  - Este estudio busca analizar alguna desigualdad en la aparición de los distintos primos que pueda ser el elemento del que se agarre algún algoritmo de machine learning o memcomputing para identificar el tipo de matrices que busquemos
- 8. Expresión en función de  $x$  e  $y$ 
  - Con este estudio, al cambiar el “espacio” de variables en el que representamos la matriz, esperamos que algún programa de machine learning o algoritmo de memcomputing sea capaz de encontrar patrones que de otra manera estarían ocultos
- 9. Estudio de 8 matrices de primos y un pseudo-primo
  - Para identificar las matrices de primos, es interesante que el programa identifique también matrices que no sean solo de primos precisamente.
- 10. Estudio de matrices de primos y no primos según número de primos
  - No solo es interesante que identifique las matrices de un número compuesto, sino que también es de gran ayuda que el programa pueda identificar propiedades en matrices con distintas cantidades de primos. Por ejemplo, en relación con la posición que ocupan
- 12. Estudio matrices de “súperprimos”
  - La idea es que estas matrices, al tener una relación “doble” con los primos podrían poseer relaciones más fuertes con el concepto de primo que facilitarían al programa su caracterización.
- 13. Estudio matrices simplificables
  - Esta clasificación sencillamente nos ayuda a identificar si hay matrices de números compuestos que, de alguna manera, podrían estar realmente compuestas por primos. Esto podría ser clave en el proceso de aprendizaje del programa.
- 14. Estudio volumen pares
  - El volumen de pares aparecido en las matrices, del mismo modo, puede ser una de las características clave que ayude al programa a identificarlas.