

Sistemas de Inteligencia Artificial

TP3: Aprendizaje No Supervisado

Grupo 6

- Desiree Melisa Limachi
- Joseph Rouquette
- Magdalena Flores Levalle
- Matias Ezequiel Daneri

Europa

Datos :

El conjunto de datos '*europa.csv*' corresponde a características económicas, sociales y geográficas de 28 países de Europa (***sin Francia*** 🙄):

Las características son :

- Country
- Area
- GDP
- Inflacion
- Life expect
- Military
- Population growth
- Unemployment

Red de Kohonen

Proyección de datos

Proyecta datos multidimensionales en una rejilla bidimensional, preservando las relaciones de los datos originales.

Actualización de pesos

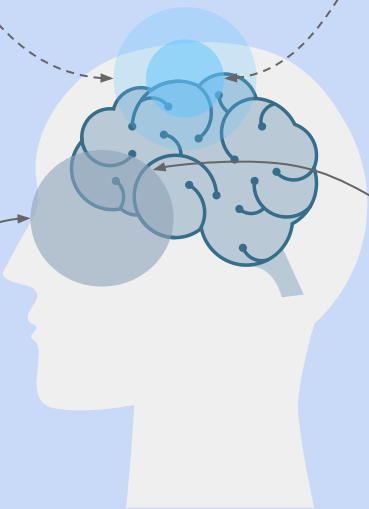
Similitud euclidiana o exponencial, para actualizar los pesos analizando sus respuestas

Iniciación de datos

*Las neuronas se inicializan con pesos **aleatorios** ajustan durante el aprendizaje para representar las características de los datos. Para el análisis comparamos con valores de entrada.*

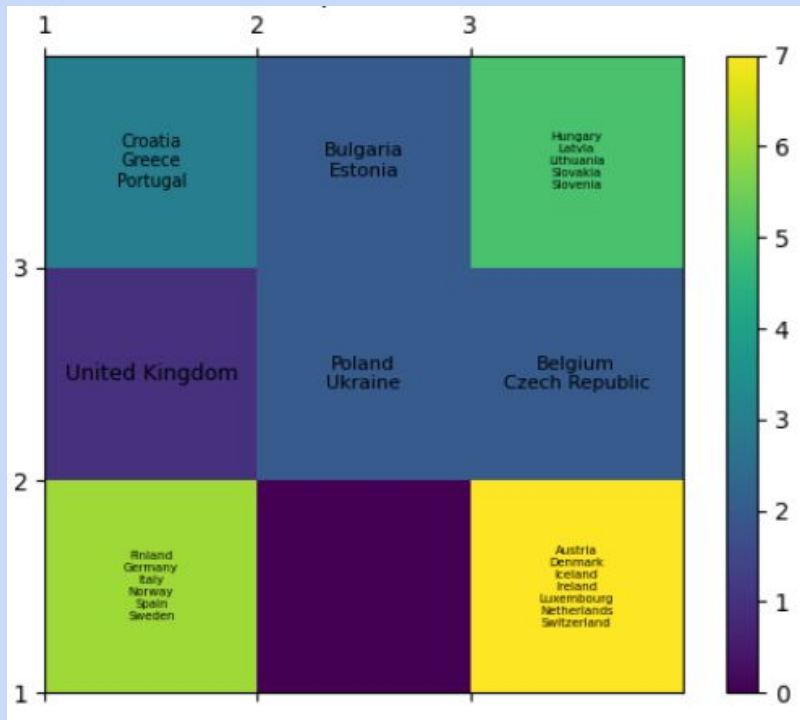
Estandarización de datos

Se utilizando la media y la desviación estándar, de modo que los datos tienen una media de cero y una desviación estándar de uno



HEATMAP 3x3

Similitud euclidiana



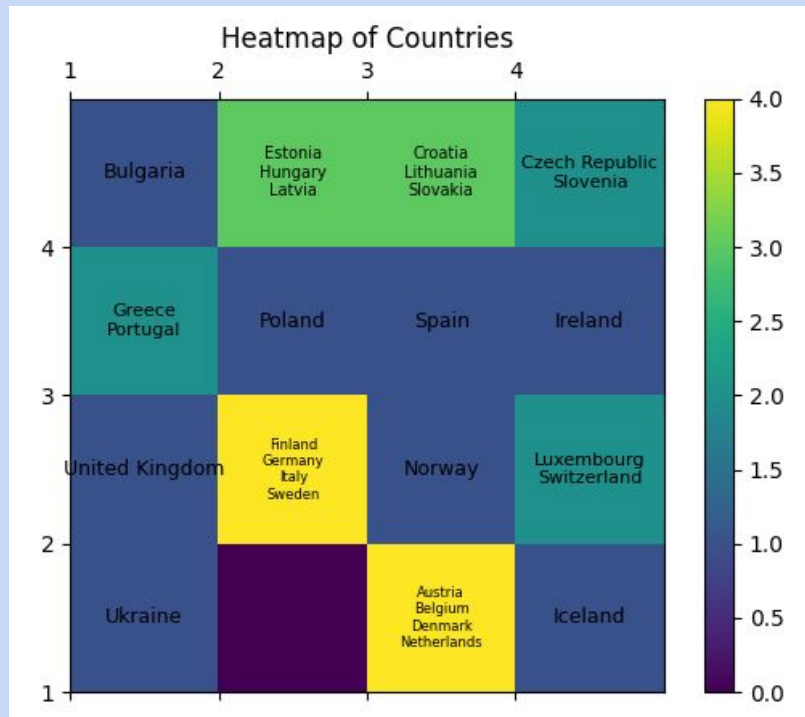
Similitud exponencial



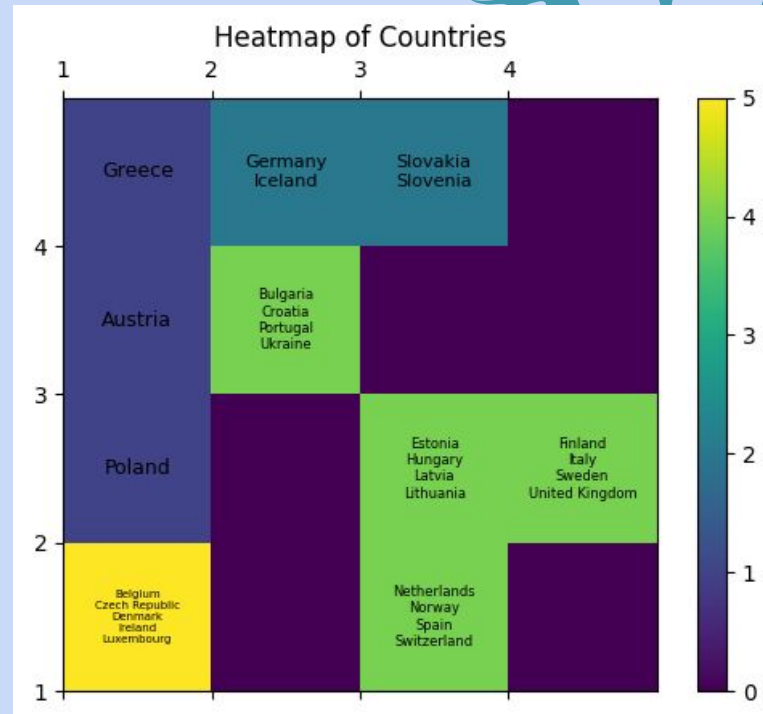
Epoch = $500 \times N = 3500$, $n = 0.01$, $R = 3$

HEATMAP 4x4

Similitud euclidean



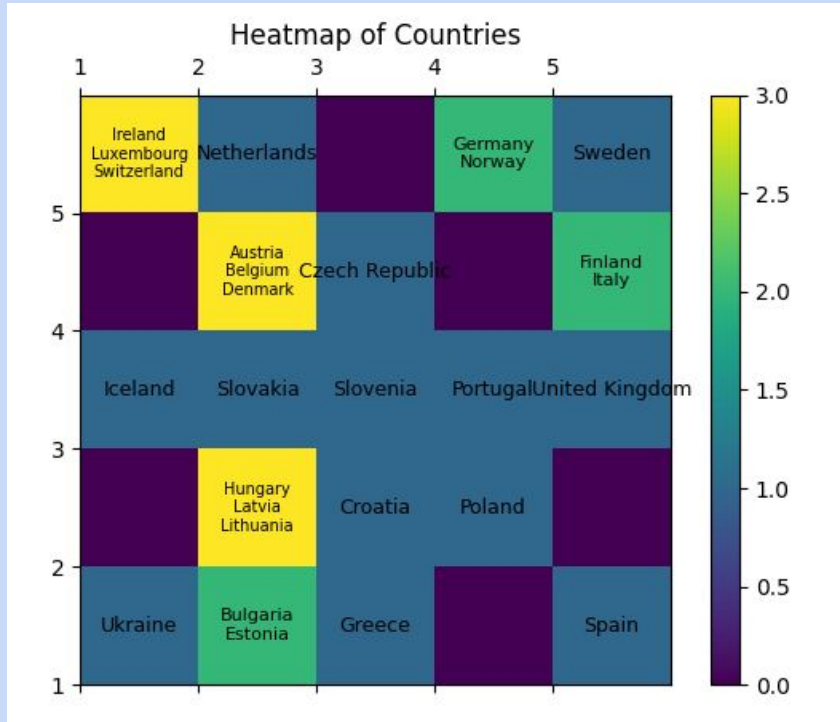
Similitud exp



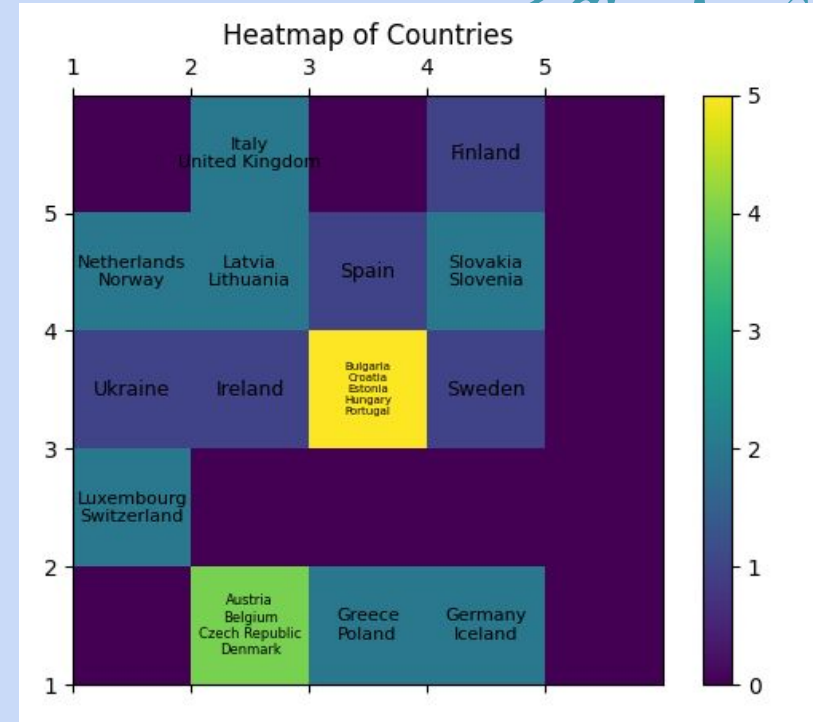
Epoch = $500 \cdot N = 3500$, $n = 0.01$, $R = 4$

HEATMAP 5x5

Similitud euclidiana



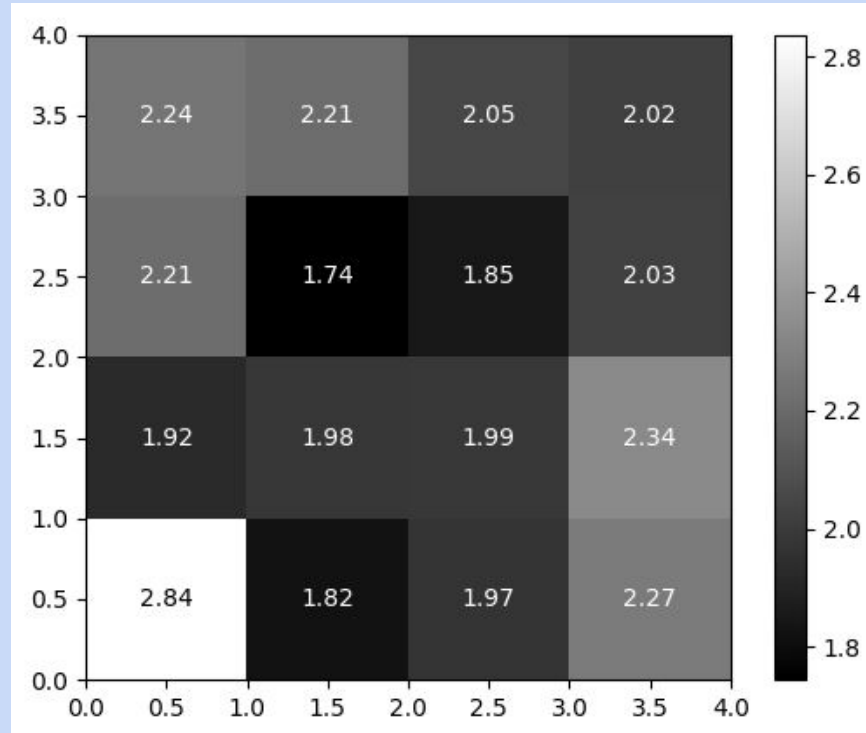
Similitud exponencial



Epoch = 500*N = 3500, n = 0.01, R = 5

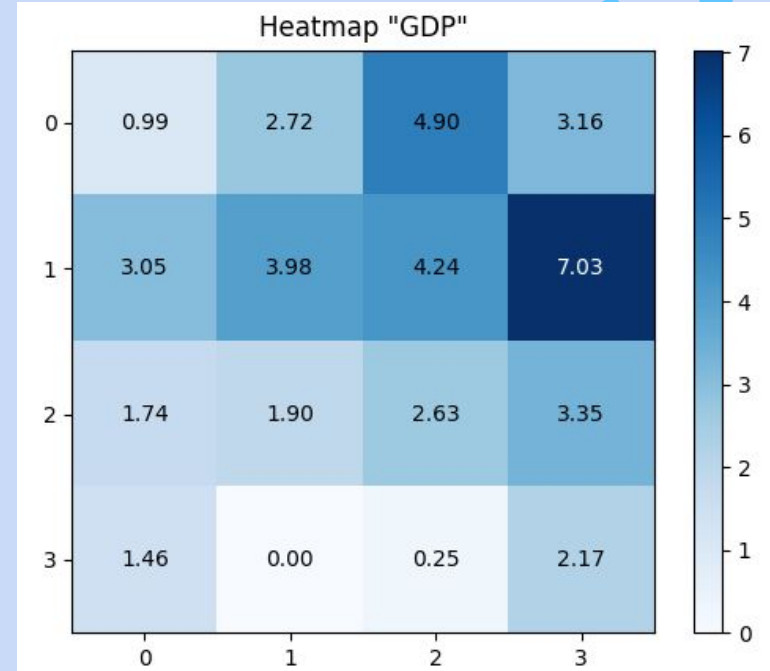
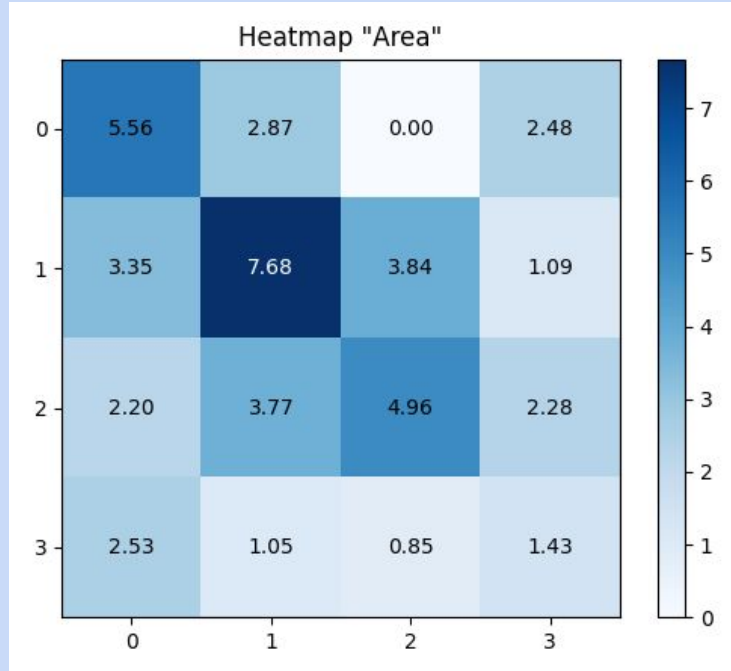
MATRIZ U DE DISTANCIA PROMEDIO

4x4

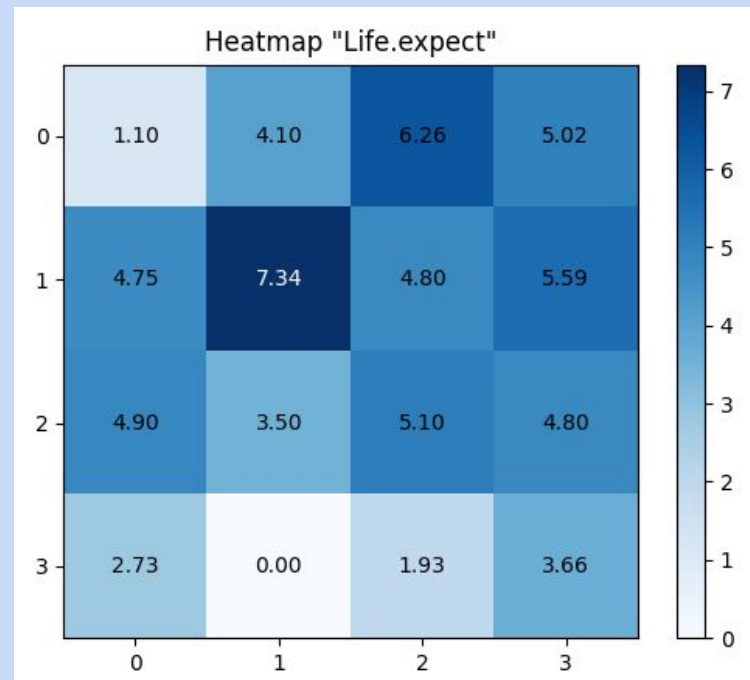
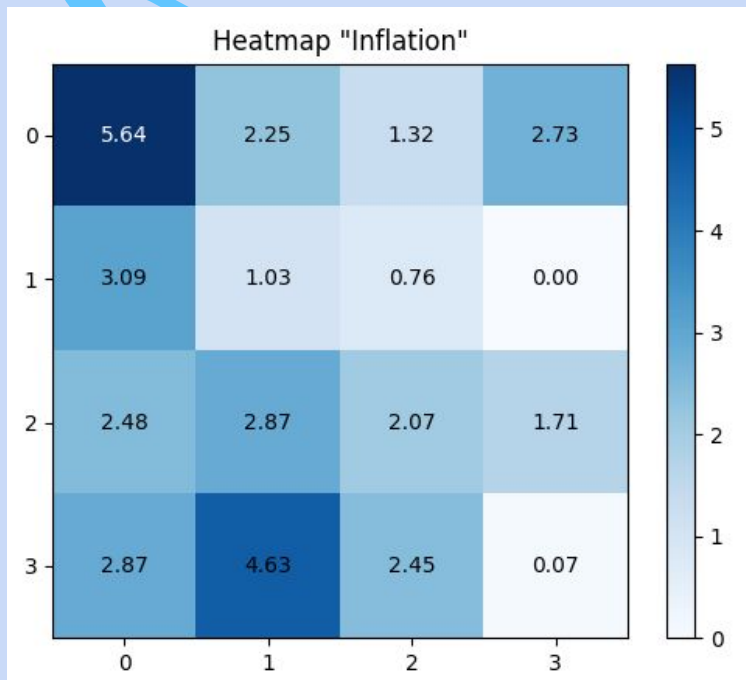


Epoch = $500 \times N = 3500$, $n = 0.01$, $R = 4$

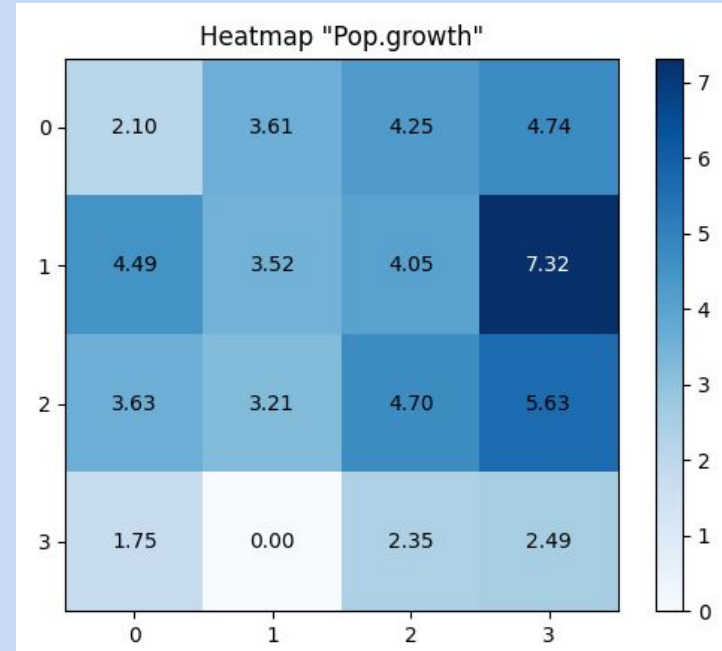
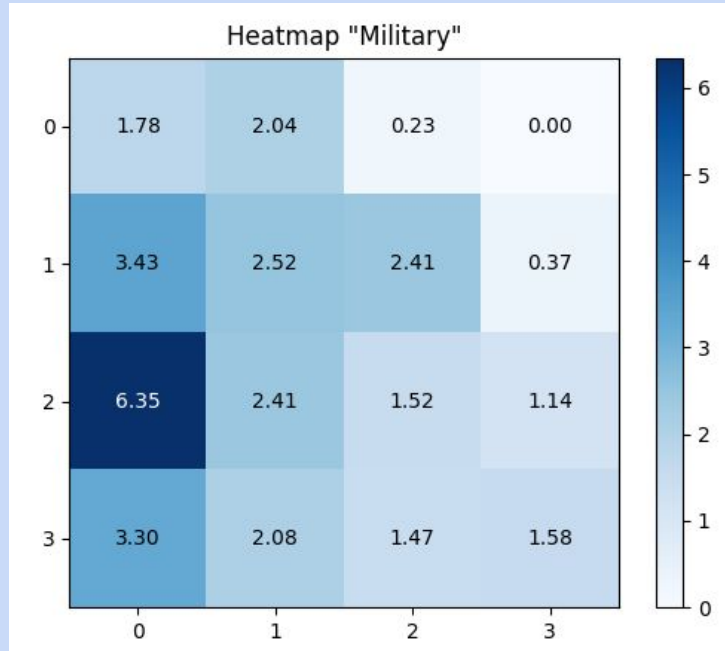
HEATMAP POR VARIABLE



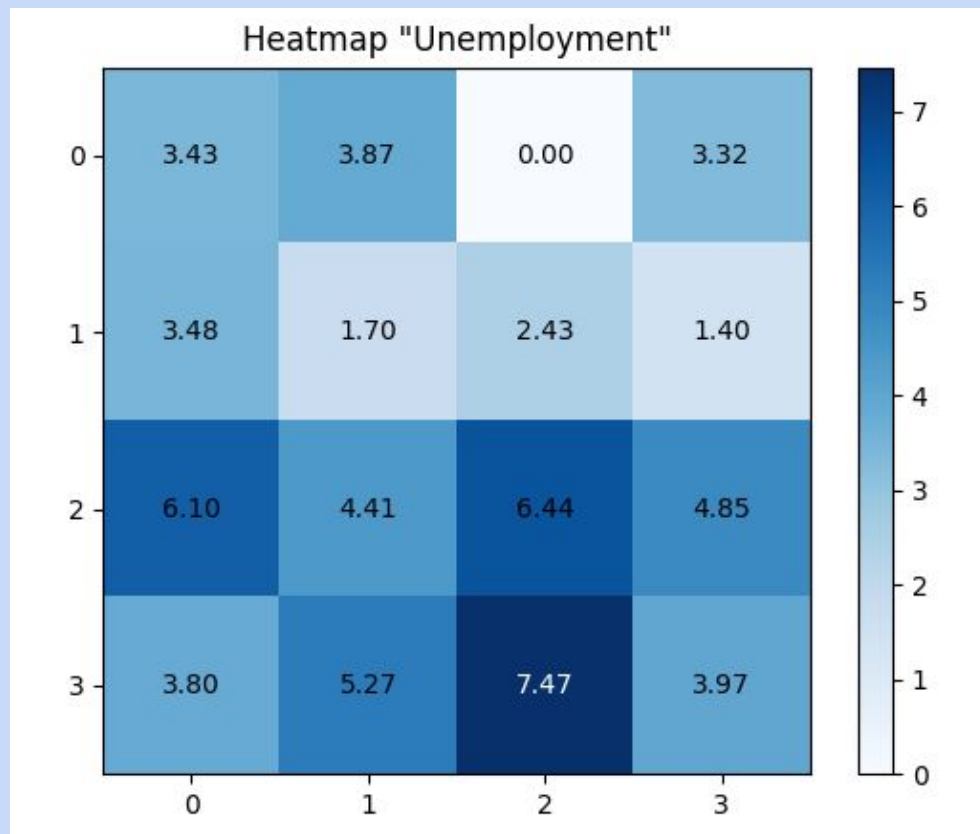
HEATMAP POR VARIABLE



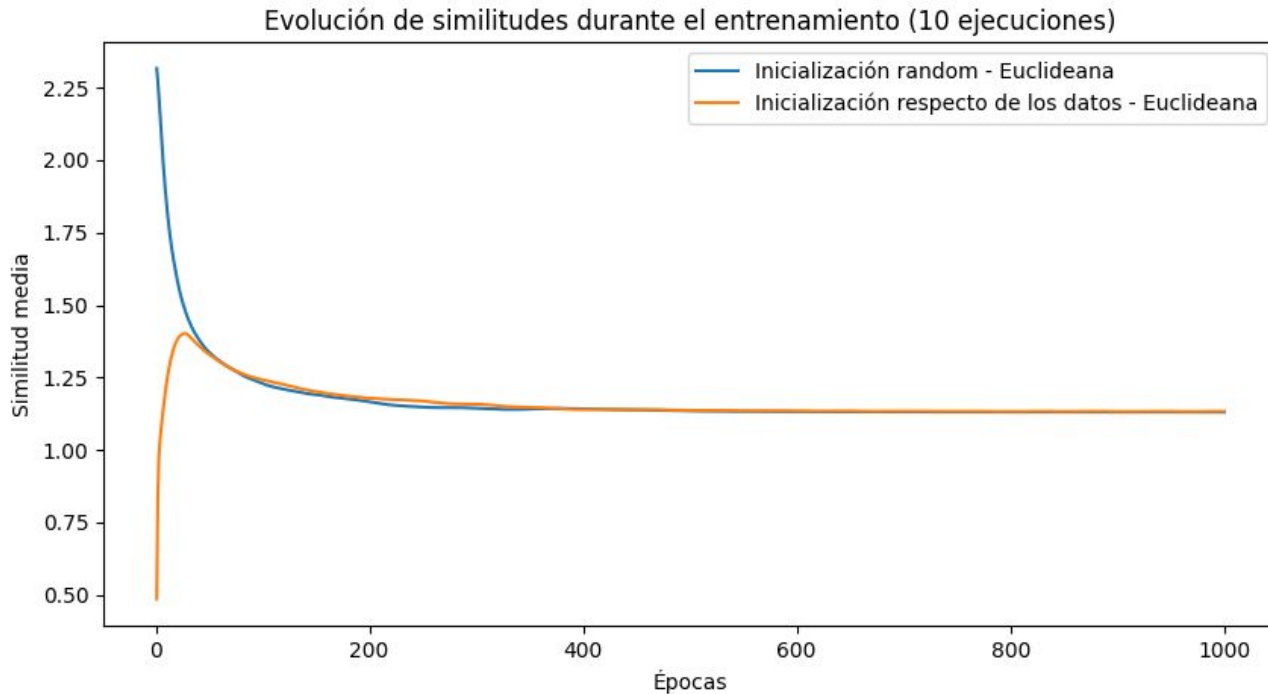
HEATMAP POR VARIABLE



HEATMAP POR VARIABLE



Inicializar los pesos aleatorios vs valores de entrada



Conclusiones



01

A mayor K , podemos encontrar un mayor número de neuronas muertas, pero podría perderse casos bordes

02

Si las distancias euclídeas entre una neurona y sus vecinos son pequeñas, indica que los vectores de pesos son similares entre sí. Por lo que al achicar el K podemos ver como van convergiendo en una neurona

03

Podemos ver una relación entre las variables “desempleo”, “crecimiento poblacional” y “guerra”, teniendo al mismo tiempo una menor expectativa de vida

Regla de Oja

Tres puntos a resolver

01

Componentes principales

Buscamos obtener la primer componente principal del conjunto de datos.

03

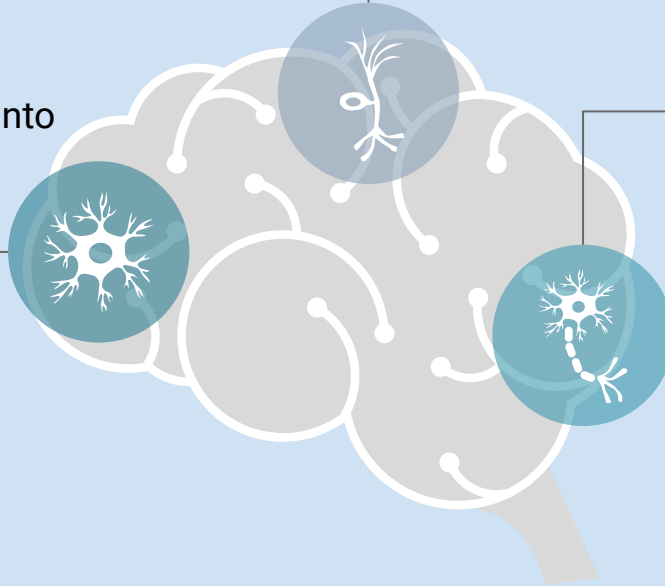
Comparar el resultado

Respecto a aplicar PCA mediante una librería.

02

Interpretar el resultado

Analizamos los pesos obtenidos.



Regla de Oja

Perceptrón Simple



La regla de Oja se basa en un perceptrón simple con una única modificación.

Actualización de los pesos



$$\Delta w = \eta (Ox_i^n - O^2 w_i^n)$$

Tasa de aprendizaje



Utilizamos una tasa de aprendizaje fija.

Componentes principales

01

Carga de datos

Cargamos los datos del archivo brindado 'europe.csv'.

02

Estandarización

Aplicamos diferencia de la media y dividimos por el desvío estándar sobre el conjunto.

03

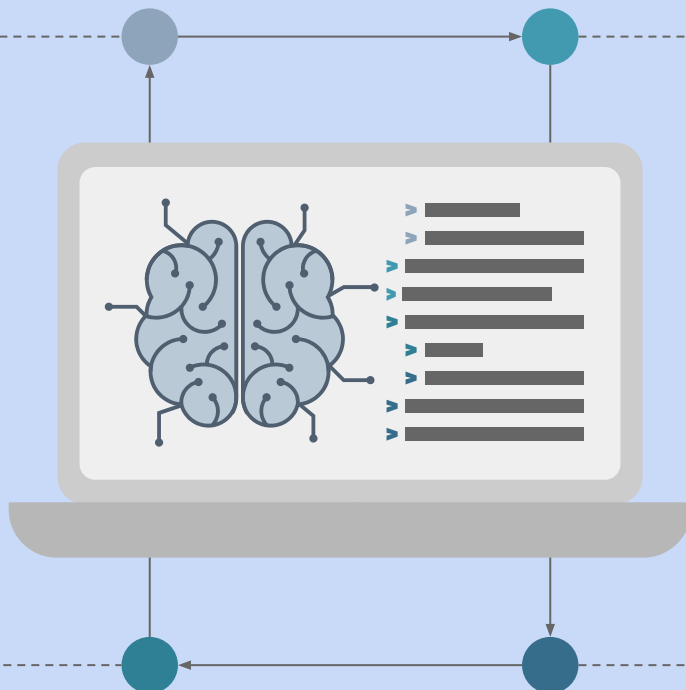
Ejecutamos Oja

Se entrena con cada fila del conjunto, exceptuando la columna *Countries*.

04

Pesos

Los pesos obtenidos, equivalen a las cargas de la primera componente principal.



Interpretación de los resultados

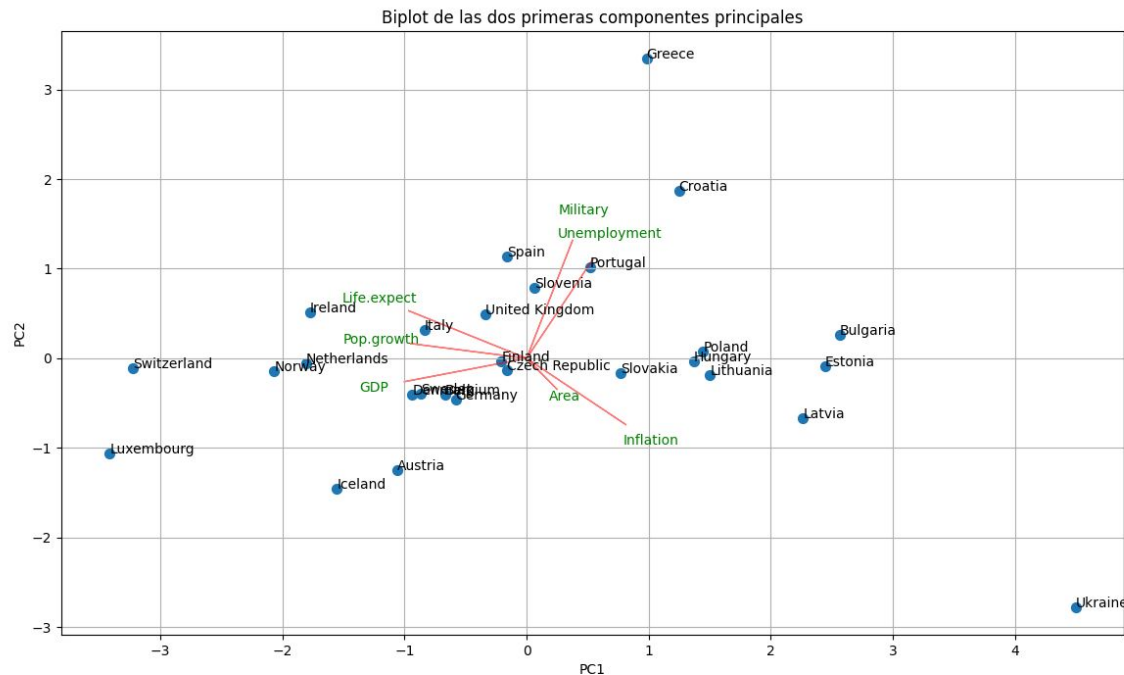
- Area: -0.13
- GDP: 0.49
- Inflación: -0.41
- Exp. de vida: 0.48
- Militares: -0.18
- Inc. de población: 0.47
- Desempleo: -0.26

Parámetros:

- Learning_rate = 0.001
- Epochs = 4000

Si bien se nos indica cada característica con signo de carga opuesto, como ya sabemos esto no indica que sea bueno o malo. Mas bien, que son características que, mientras una incrementa la otra decrementará.

Recordemos: Biplot con Sklearn y PCA

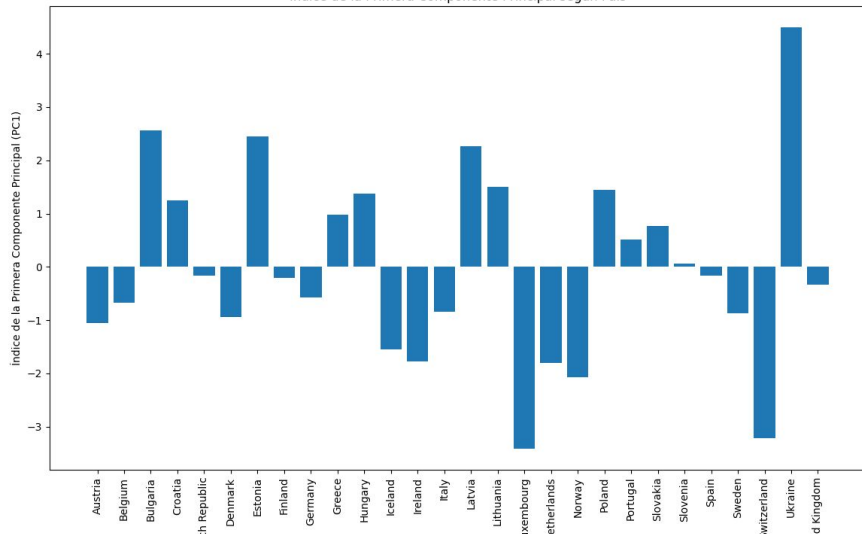


El PC1 dividió a los países en dos grupos :

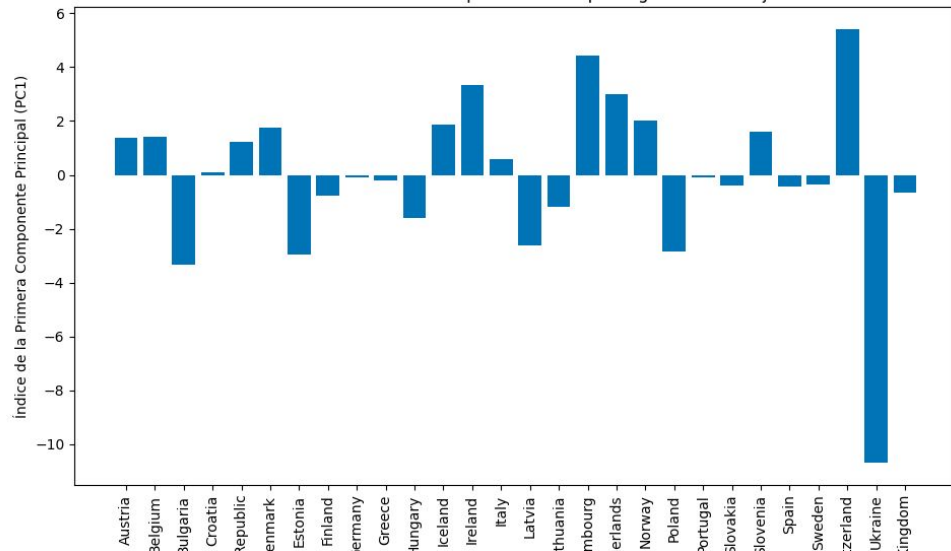
- Military, Unemployment, Inflation, Area
- Life.expect, Pop.growth, GDP

Comparación de PC1 con Sklearn y Oja

Índice de la Primera Componente Principal según País



Índice de la Primera Componente Principal según País con Oja

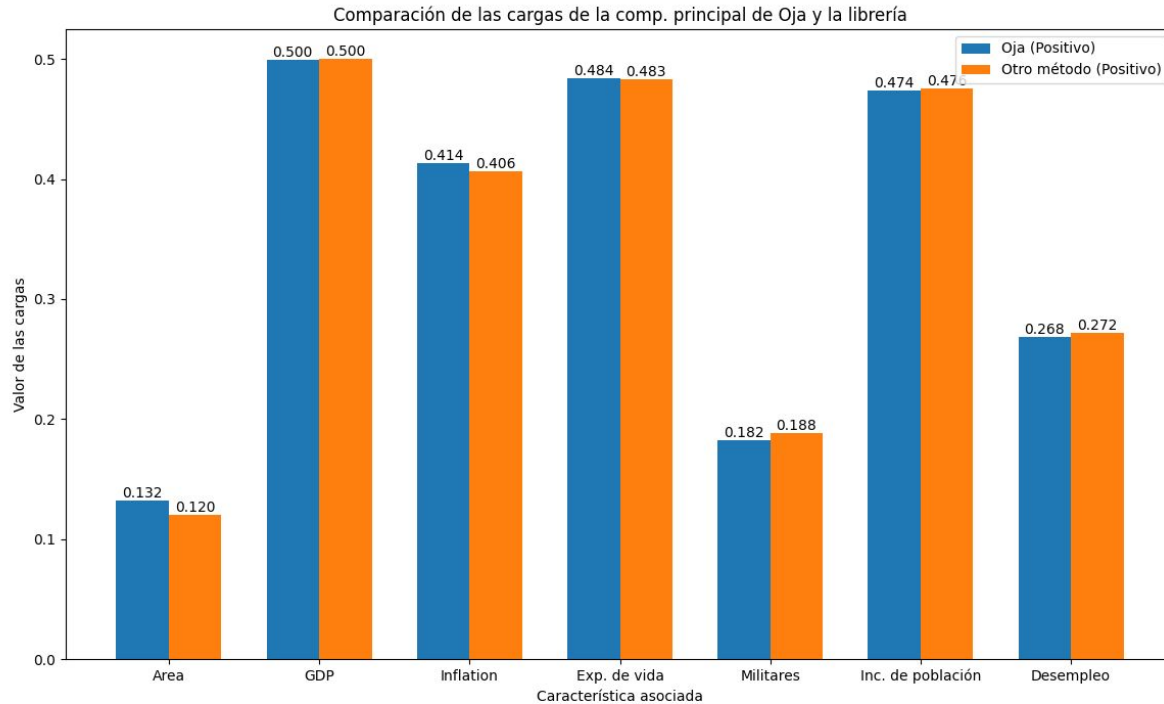


Con un learning rate de 0.1, obtuvimos un error total de **0.9465**

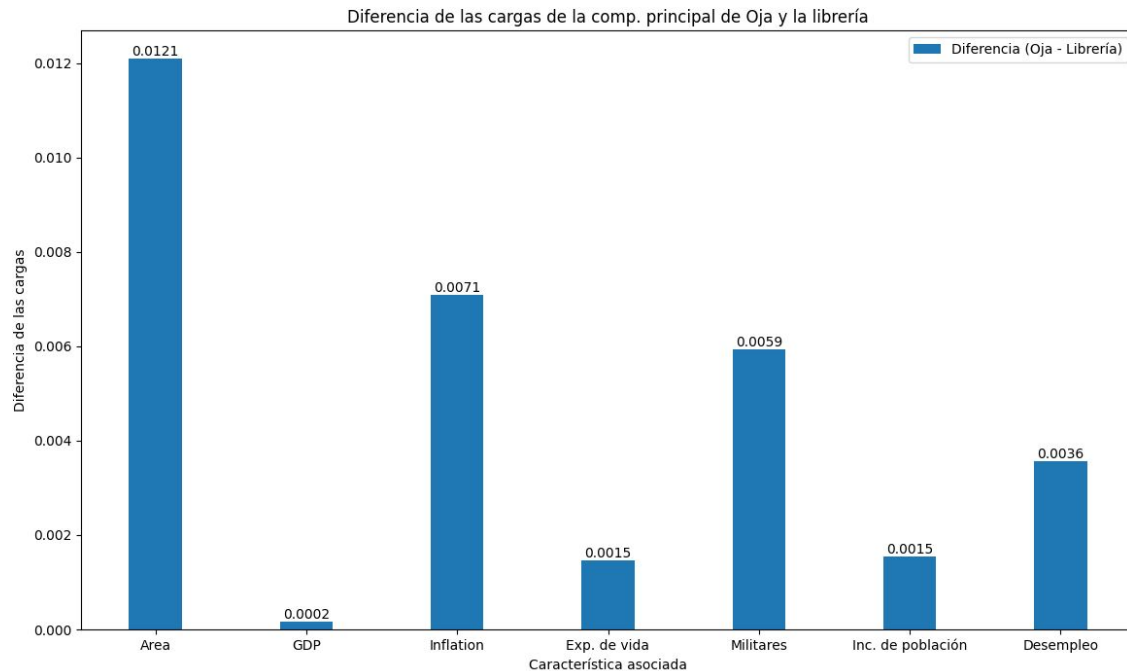
Comparación de resultados

Parámetros:

- Learning_rate = 0.001
- Epochs = 4000



Comparación de resultados



El error total es **0.0319**

Conclusiones

A diagram of a neuron with a central blue nucleus and branching dendrites. Two callout boxes, labeled '01' and '02', are connected to the neuron by lines. Box '01' points to the dendrites, and box '02' points to the nucleus.

01

La regla de Oja es muy eficaz para aproximar la primer componente principal y se pueden obtener valores cercanos a los obtenidos mediante la librería.

02

La ventaja de utilizar la regla de Oja, es que además, se puede entrenar el modelo con nuevos datos.

- Mientras que con la librería, es decir, aplicando autovalores y autovectores, el modelo queda ajustado a los datos desde un inicio.



HOPFIELD

Algoritmo de aprendizaje no supervisado que utiliza una red neuronal recurrente para almacenar y recuperar patrones de memoria. Existe retroalimentación entre las neuronas ya que están todas conectadas entre sí

Parametros iniciales

- Creamos un red 5x5 y entrenamos para reconocer **A, F, J, X** teniendo en cuenta que son letras topológicamente bastante distintas.
- Analizaremos si nuestro modelo puede distinguir letras diferentes de las de entrenamiento, las mismas y las mismas con ruido.
- Utilizamos una actualización secuencial aleatoria, donde se modulan todas las neuronas en cada paso, pero solo se eligen 5 al azar para ser actualizadas.

Pruebas - mismas letras

```
  * * *
*       *
* * * * *
*       *
*       *

-97.5
Paso 1:
  * * *
*       *
* * * * *
*       *
*       *

-97.5
Paso 2:
  * * *
*       *
* * * * *
*       *
*       *

-97.5
```

```
  * * * * *
*
* * * *
*
*

-92.5
Paso 1:
* * * * *
*
* * * *
*
*

-92.5
Paso 2:
* * * * *
*
* * * *
*
*

-92.5
```

```
  * * * * *
      *
      *
      *
* *

-70.5
Paso 1:
* * * * *
      *
      *
      *
* *

-70.5
Paso 2:
* * * * *
      *
      *
      *
* *

-70.5
```

```
  *           *
    *       *
      *     *
    *       *
  *         *

-74.5
Paso 1:
  *           *
    *       *
      *     *
    *       *
  *         *

-74.5
Paso 2:
  *           *
    *       *
      *     *
    *       *
  *         *

-74.5
```


Pruebas - letras con ruido [10%, 20%, 30%]

```
* * * * *
*
* * * *
*
*
-85.5
Paso 1:
* * * * *
*
* * * *
*
*
-92.5
Paso 2:
* * * * *
*
* * * *
*
*
-92.5
```

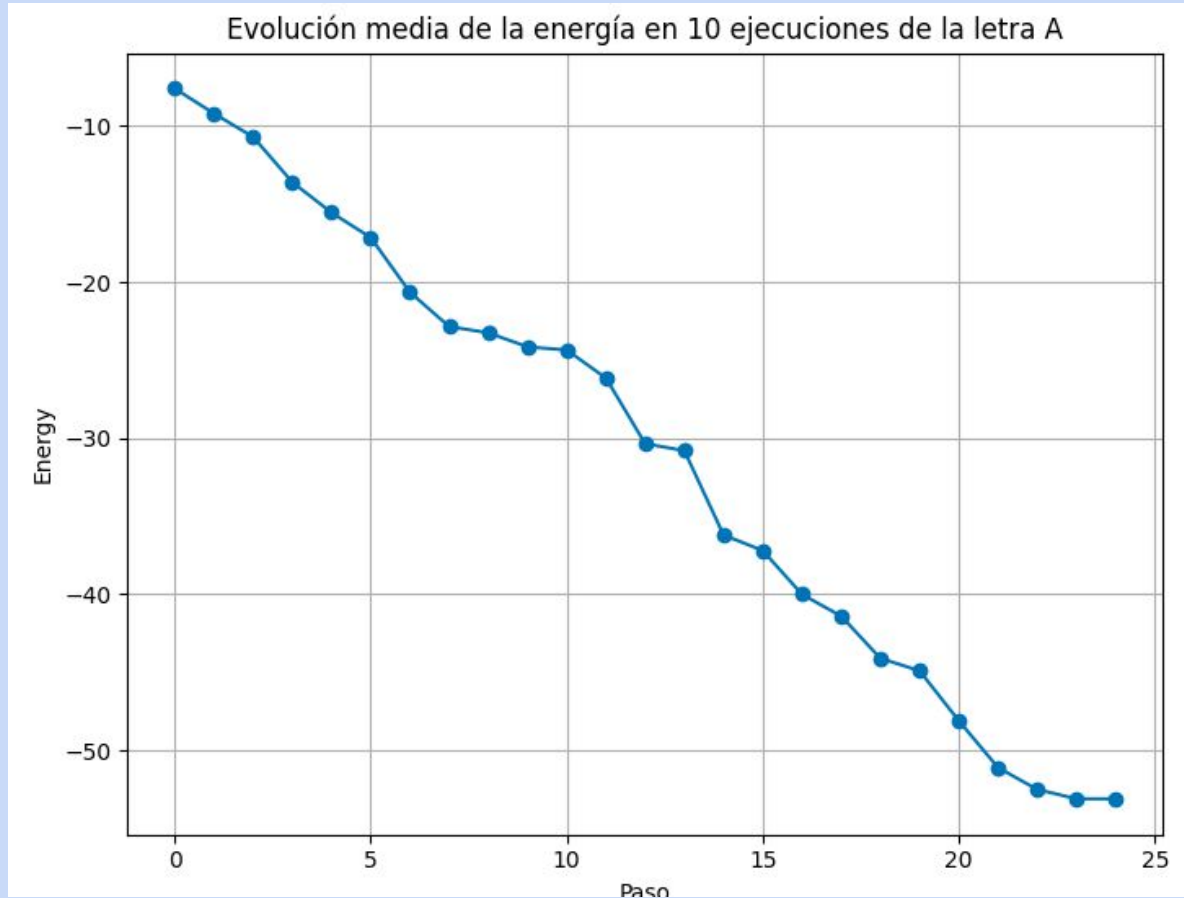
```
* * * *
* *
* * *
*
*
-45.5
Paso 1:
* * * *
* *
* * *
*
*
-45.5
Paso 2:
* * * *
* *
* * *
*
*
-45.5
```

```
Paso 3:
* * * *
* *
* * *
*
*
-45.5
Paso 4:
* * * *
*
* * *
*
*
-60.5
Paso 5:
* * * *
*
* * * *
*
*
-71.5
```

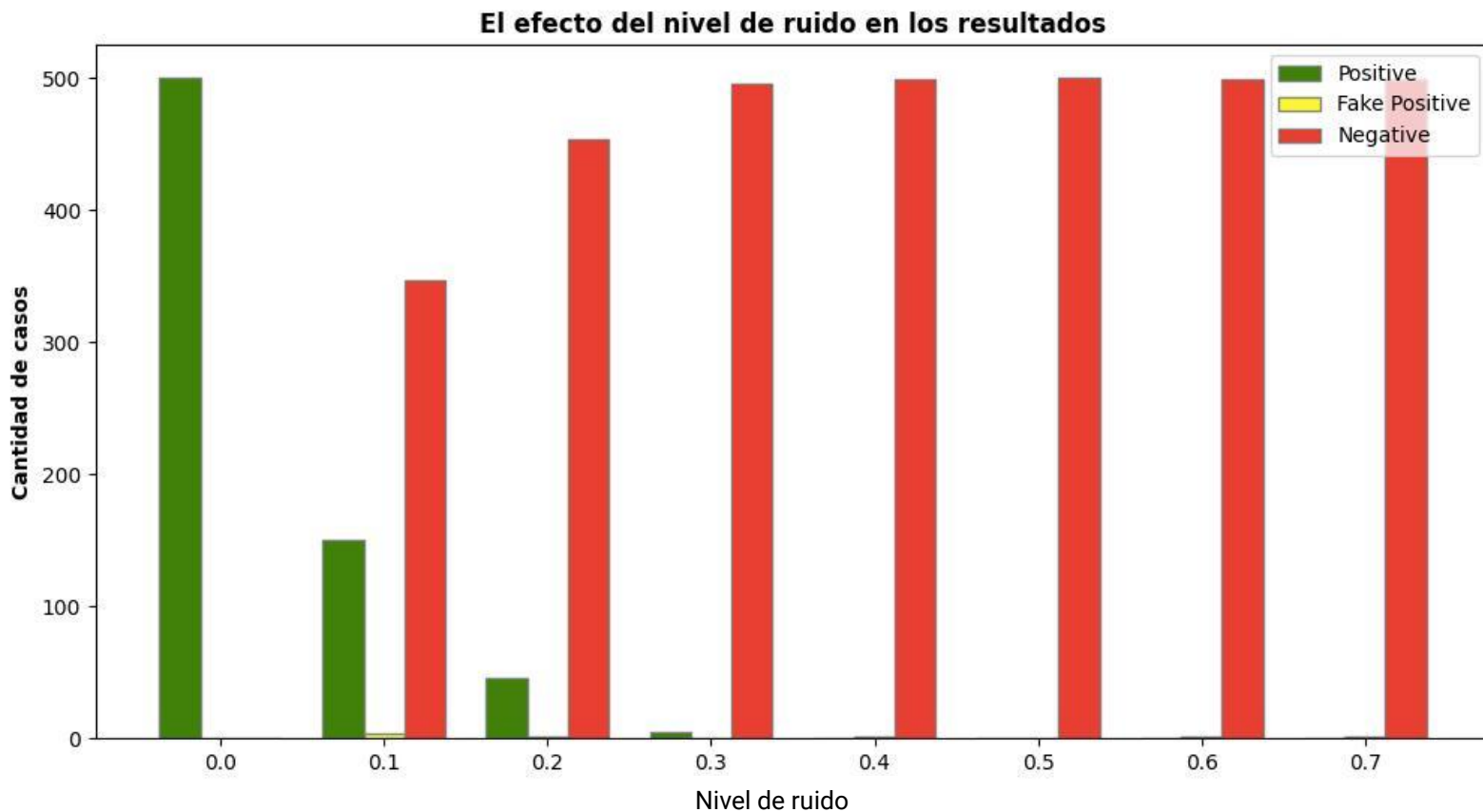
```
* * * * *
* *
*
*
-6.5
Paso 1:
* * * * *
*
* * *
*
* * *
-22.5
Paso 2:
* * * * *
*
* * *
*
* * *
-23.0
```

```
Paso 3:
* * * * *
*
* * *
*
* * *
-23.0
Paso 4:
* * * * *
*
* * *
*
* *
-26.0
Paso 5:
* * * * *
*
* * *
*
* *
-26.0
```

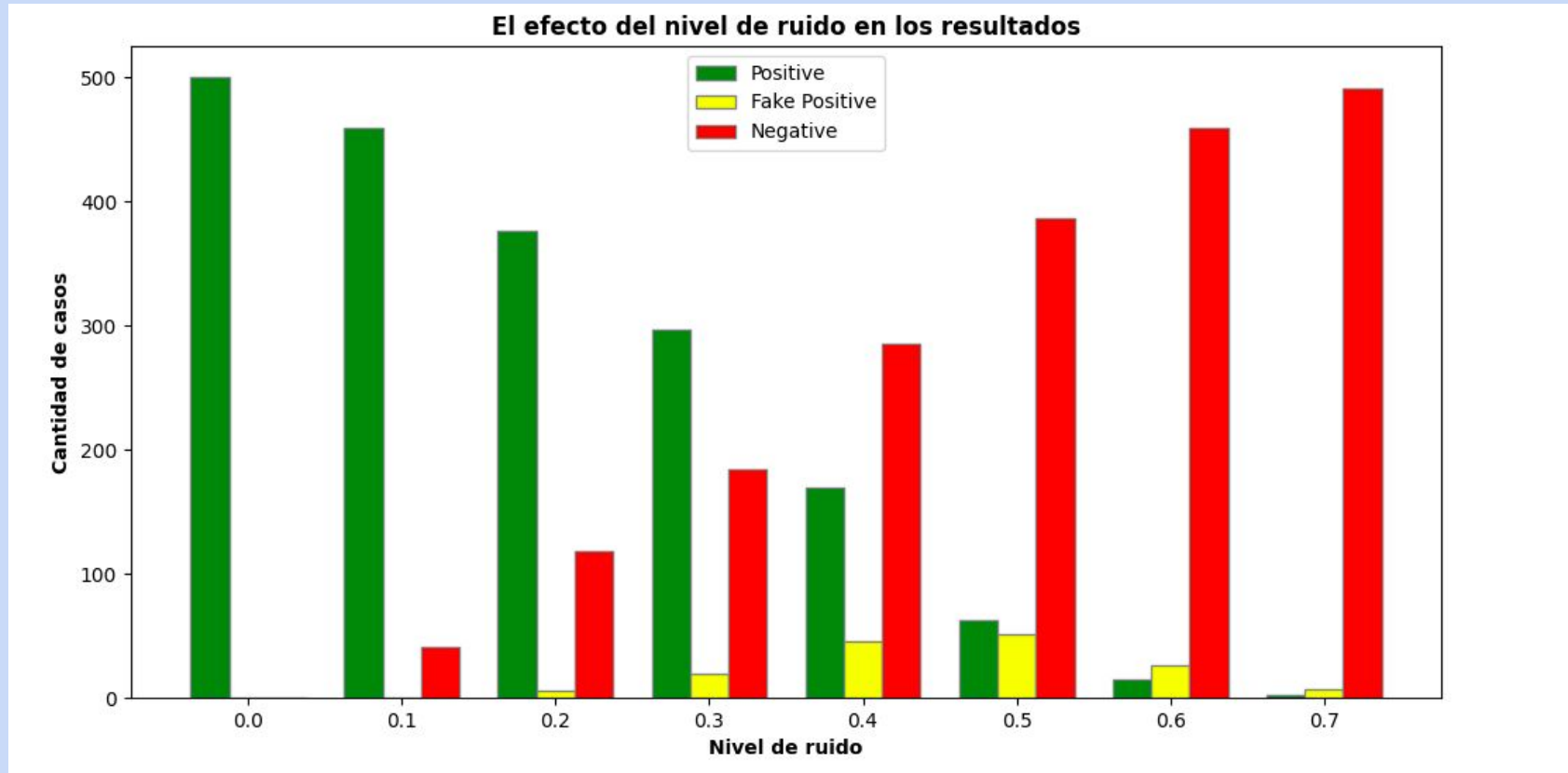
Evolución media de la energía



Efectos del ruido con 5 neuronas cambiados a cada paso



Efectos del ruido con todas las neuronas a cada paso



Conclusiones



01

Para niveles de ruido bajos a moderados, la red de Hopfield puede recuperar exitosamente el patrón original.

02

Los estados espurios se vuelven más comunes cuando el nivel de ruido es alto o cuando hay muchas similitudes entre los patrones almacenados, lo que causa confusiones en la red.

03

Es posible observar variaciones en los patrones recuperados, incluso con el mismo nivel de ruido y el mismo patrón original.



GRACIAS!