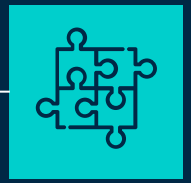
TP4: Aprendizaje no supervisado

Grupo 2
Tomás Álvarez Escalante (60127)
Alejo Francisco Caeiro (60692)
Lucas Agustín Ferreiro (61595)
Román Gómez Kiss (61003)

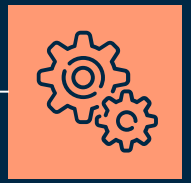
Tabla de contenidos



01

Modelo de Kohonen

Resolución del ejercicio 1.A



02

Regla de Oja

Resolución del ejercicio 1.B

03

Modelo de Hopfield

Resolución del ejercicio 2

Modelo de Kohonen

01

Problema

Se cuenta con un conjunto de datos de 28 países europeos:

- Country: nombre del país.
- Área: área geográfica.
- GDP: producto bruto interno.
- Inflation: inflación anual.
- Life.expect: expectativa de vida media en años.
- Military: nivel militar.
- Pop.growth: tasa de crecimiento poblacional.
- Unemployment: tasa de desempleo.

Queremos asociar países con cualidades sociopolíticas similares, en base a los datos otorgados.

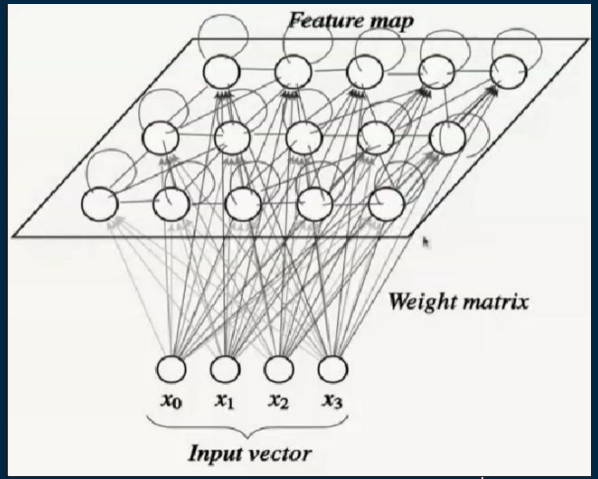
Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
Austria	83871	41600	3,5	79,91	0,8	0,03	4,2
Belgium	30528	37800	3,5	79,65	1,3	0,06	7,2
Bulgaria	110879	13800	4,2	73,84	2,6	-0,8	9,6
Croatia	56594	18000	2,3	75,99	2,39	-0,09	17,7
Czech Republic	78867	27100	1,9	77,38	1,15	-0,13	8,5
Denmark	43094	37000	2,8	78,78	1,3	0,24	6,1

Red de Kohonen

Utilizaremos la Red de Kohonen, que es una red de una sola capa en forma de grilla bidimensional donde podremos agrupar a los países por sus características.

El objetivo es agrupar los datos para que las informaciones similares sean clasificadas formando parte de una misma categoría, es decir, activando una misma neurona del mapa o capa de salida.

Se inicializan los pesos con muestras de los datos de entrada para evitar que algunas unidades queden lejos de los valores iniciales y tener neuronas muertas.



Funcionamiento de Kohonen

Se selecciona un registro de entrada Xp y se selecciona la neurona ganadora que tenga el vector de pesos más cercano a Xp mediante un método de similitud:

Distancia euclidea

$$W_{\hat{k}} = 1 \leq \underset{j \leq N}{\arg\min} \{d(X^P - W_j)\}$$

Exponencial

$$W_{\hat{k}} = 1 \leq \underset{j \leq N}{\arg\min} \{e^{-||X^P - W_j||^2}\}$$

Luego se actualizan los pesos del vecindario de la neurona ganadora según la ***regla de Kohonen***:

Vecindario: $N_{\hat{k}}(i) = \{n \text{ tal que } ||n - n_k|| < R(i)\}$

a. Si $j \in N_{\hat{k}}(i) \rightarrow W_j^{i+1} = W_j^i + \eta(i) * (X^P - W_j^i)$

b. Si no son vecinas, es decir, $j \notin N_{\hat{k}}(i) \rightarrow W_j^{i+1} = W_j^i$

En cada iteración se actualizan η(i) y R(i) con la siguiente regla: $\eta(i) = \eta(0) * e^{-i/\text{Total iterations}}$

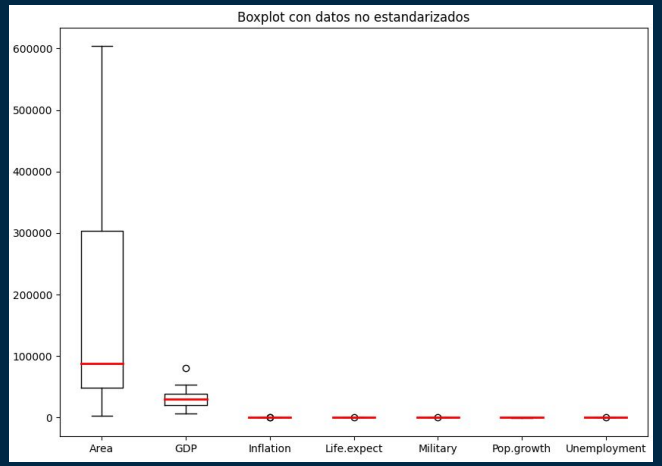
Consideraciones de estandarización

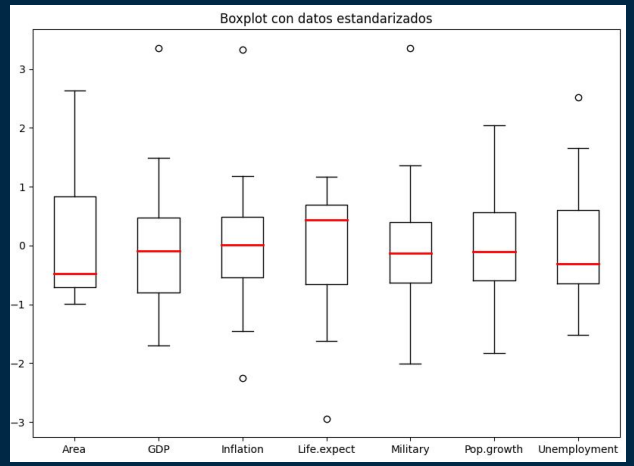
Para que la red funcione correctamente, es necesario estandarizar el conjunto de datos, ya que si algunas de las variables toman valores mayores a los demás, estas tendrán mayor varianza, pero no necesariamente mayor variabilidad, y además las variables con valores más grandes tendrán más peso en el análisis.

$$\tilde{X}_i = \frac{X_i - \bar{X}_i}{s_i}$$

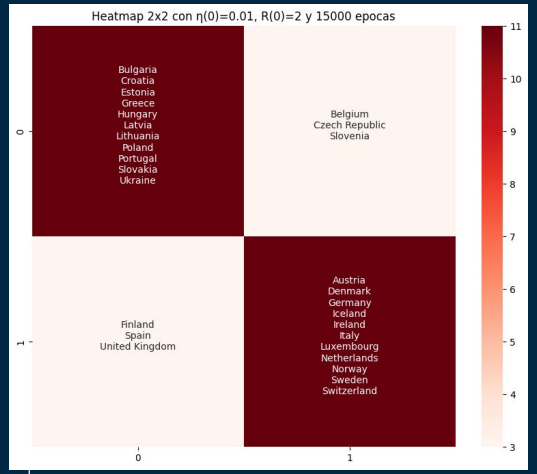
$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_i^j$$

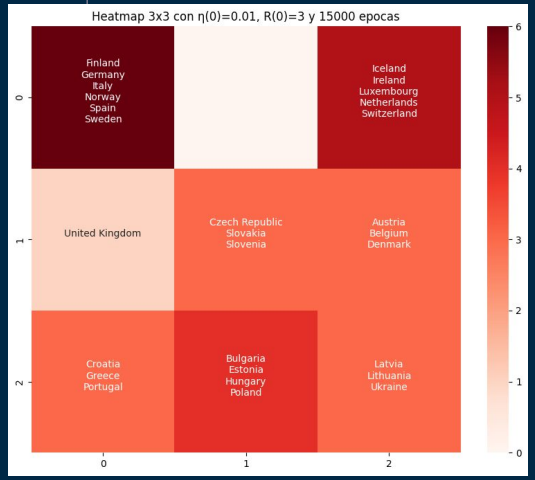
$$s_i = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(X_i^j - \bar{X}_i)}$$

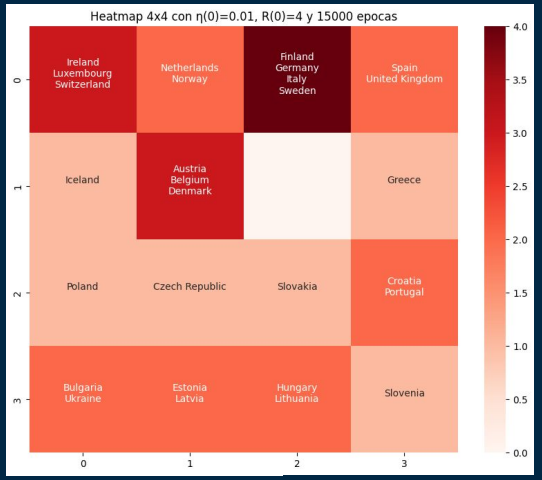


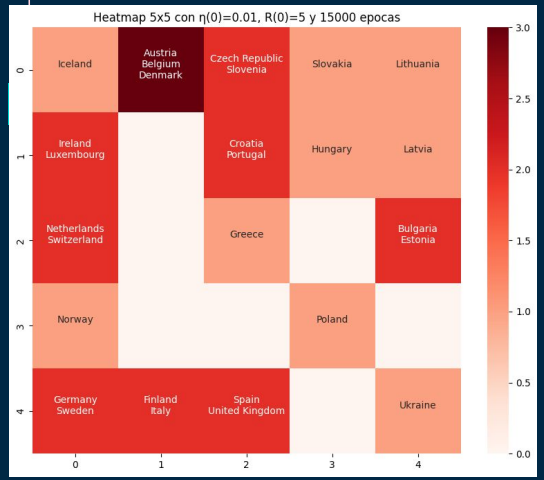


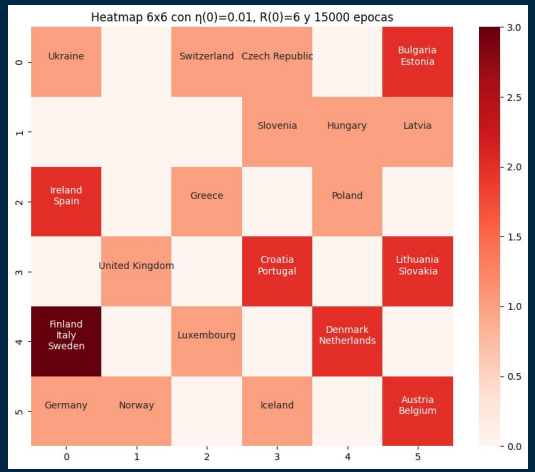
Variando distintos tamaños de grilla







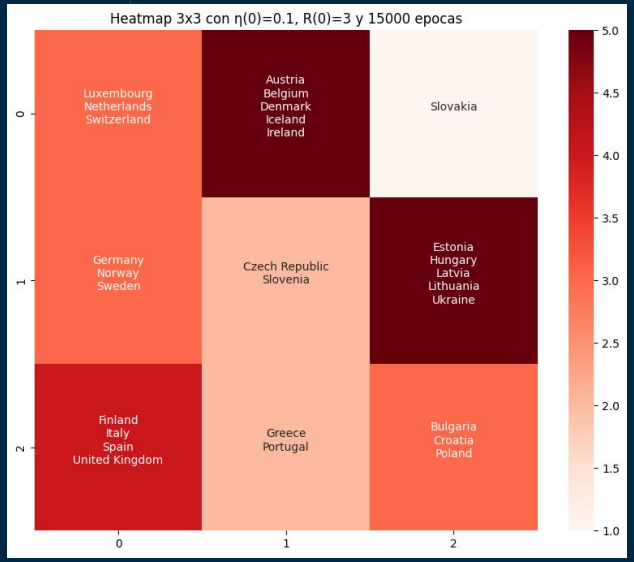


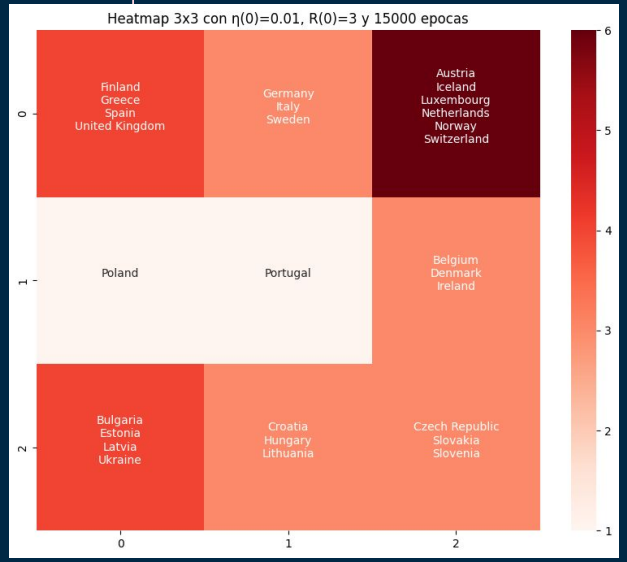


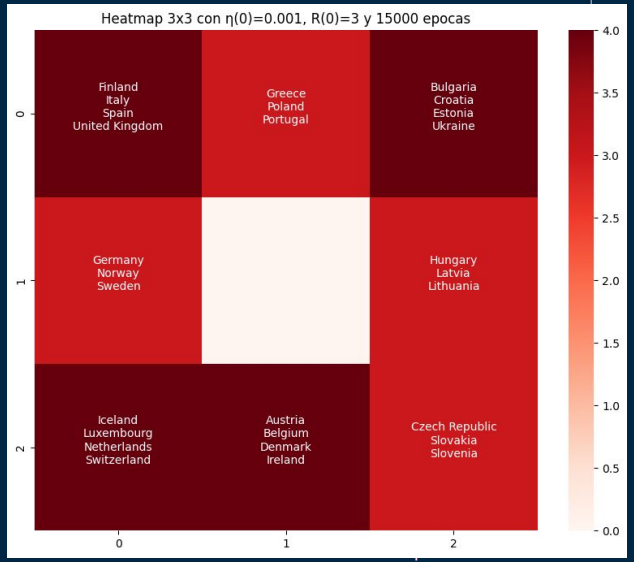
Podemos ver que a un valor de K mayor, se podrá hacer una clasificación más fina de los países, a coste de tener más neuronas muertas. Puede llegar a suceder que si tenemos un alto valor de η o épocas, queden países en grupos de 1.

Resultados obtenidos con K=3

Variando la η

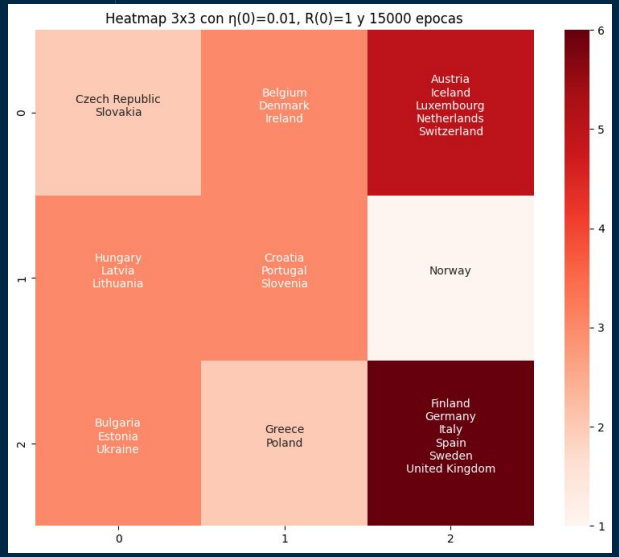


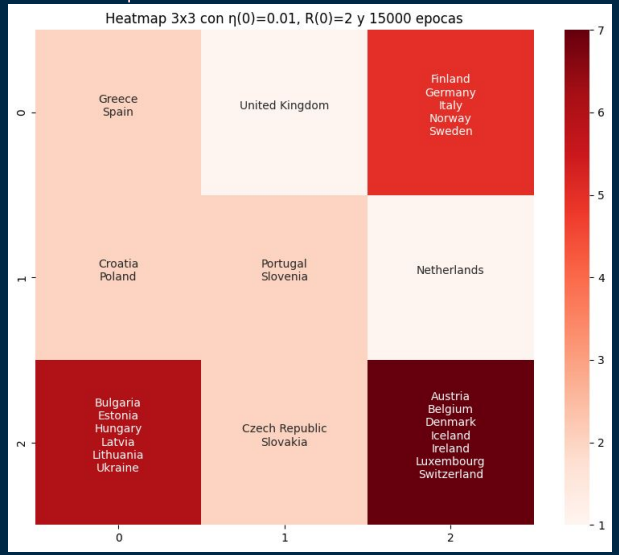


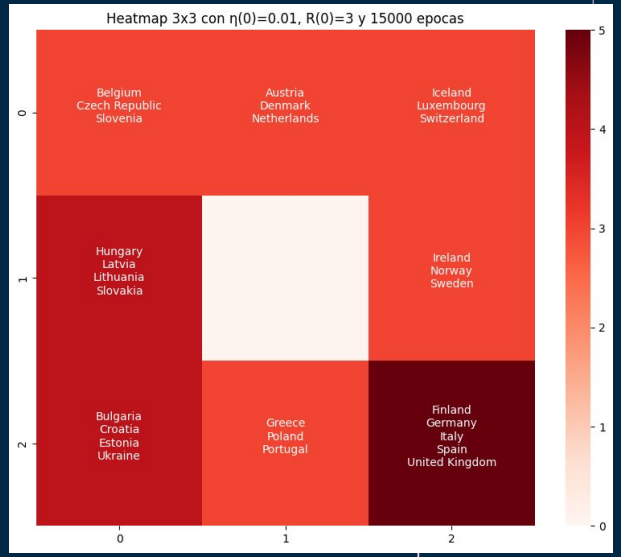


Cuanto menor valor de η, se convergerá a una solución donde los países quedan más agrupados.

Variando el radio

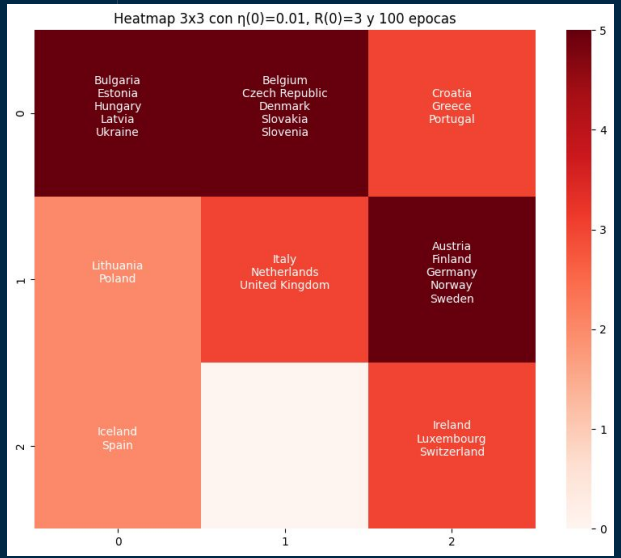


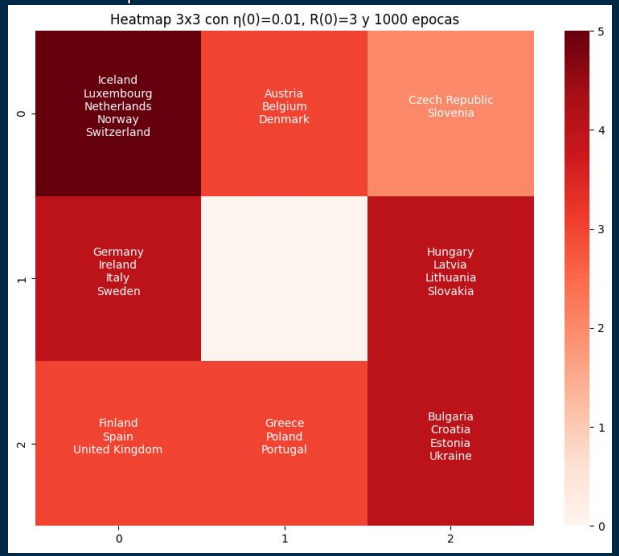


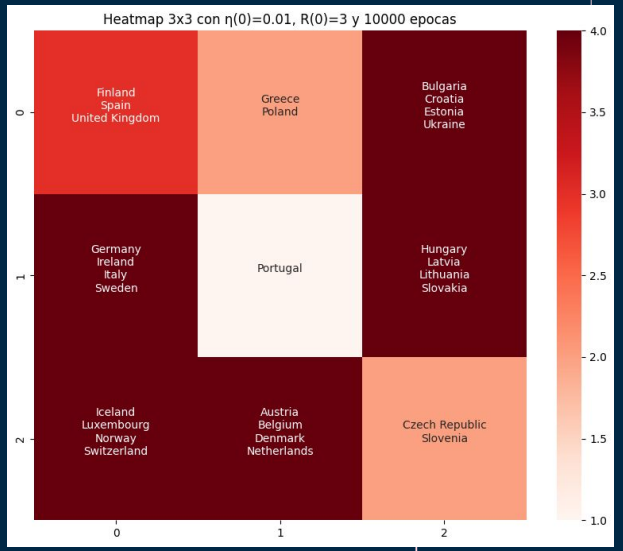


A mayor radio los cambios se distribuirán más en la red, modificando los pesos en mayor medida en cada iteración. Esto llevará a mayor precisión y una convergencia más rápida

Variando las épocas

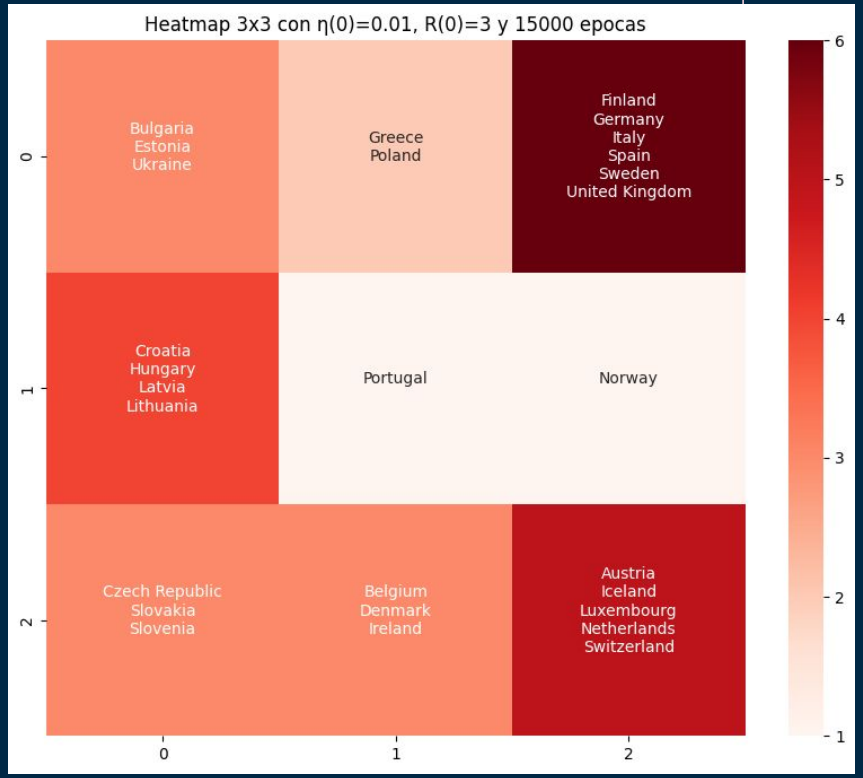


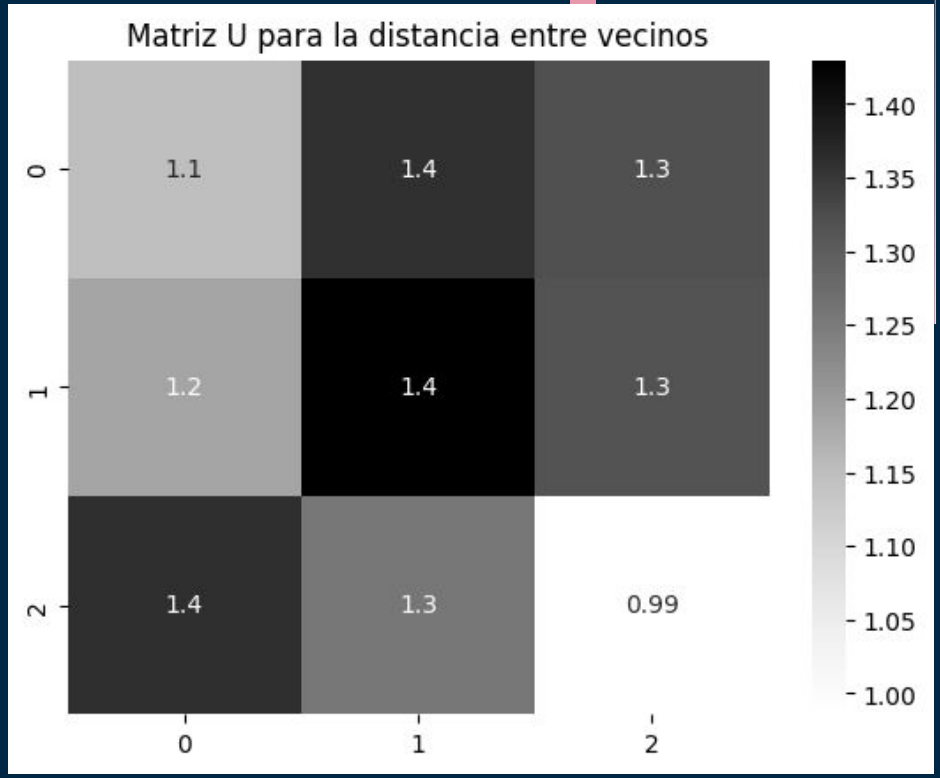




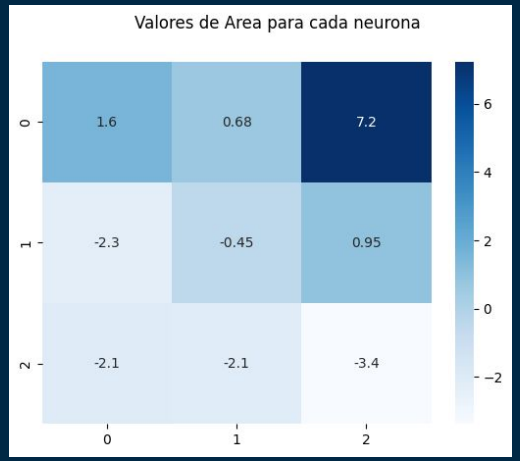
Al tener un valor más grande de cantidad de épocas, los países quedarán más distribuidos, ya que la red tendrá más tiempo para “aprender” como clasificarlos.

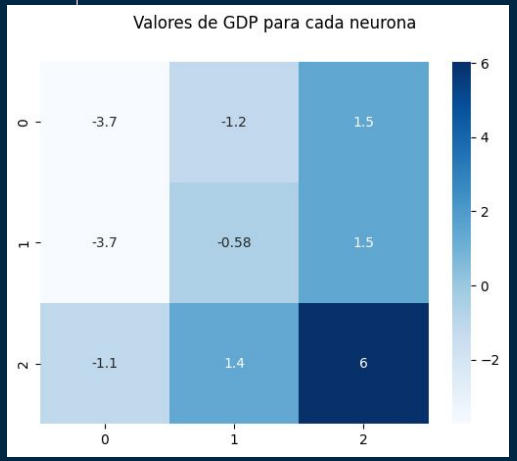
Heatmap y Matriz U

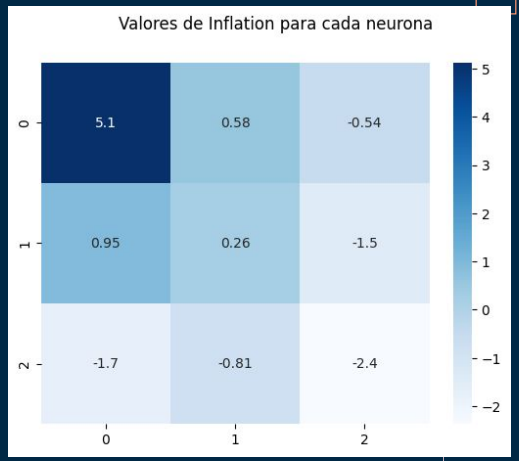


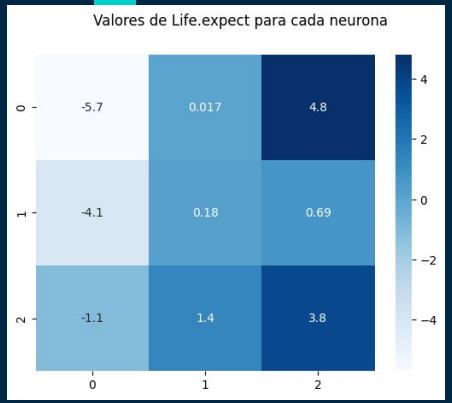


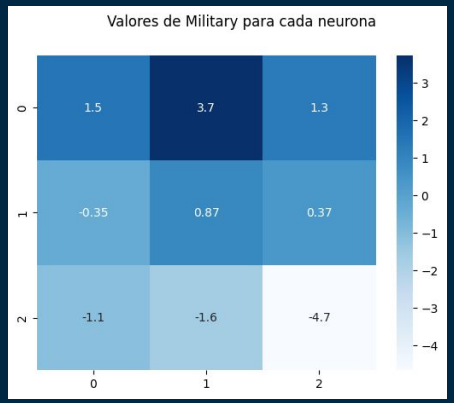
Heatmap de una sola variable

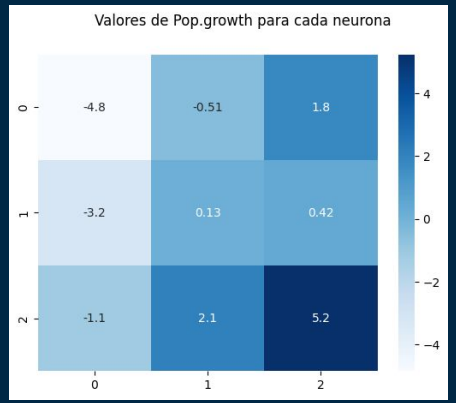


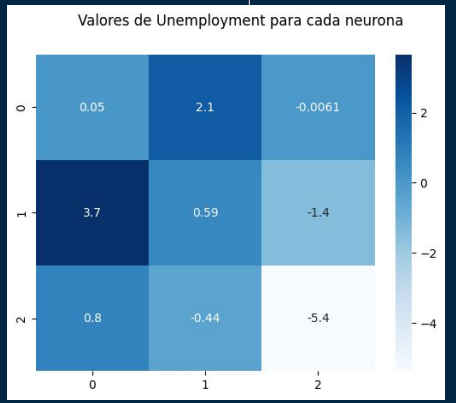






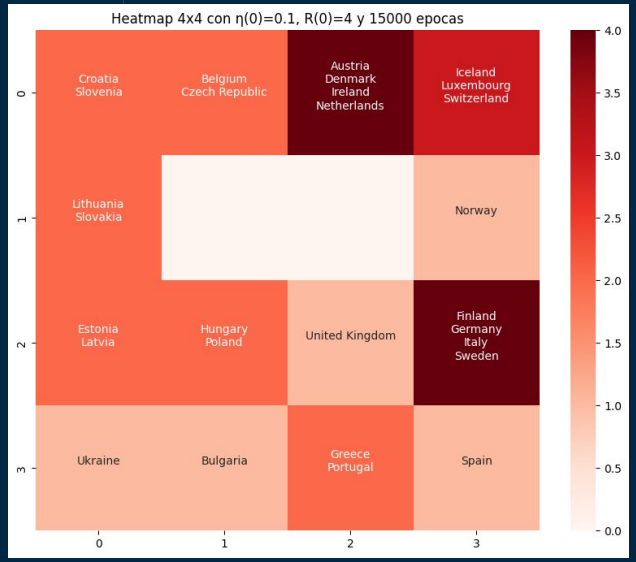


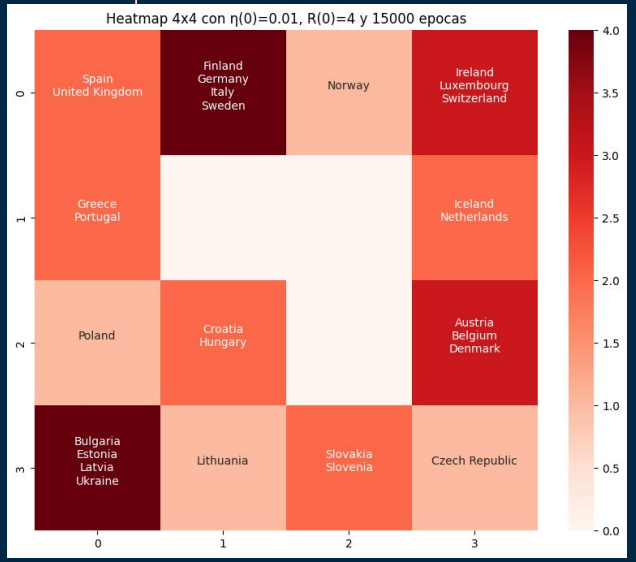


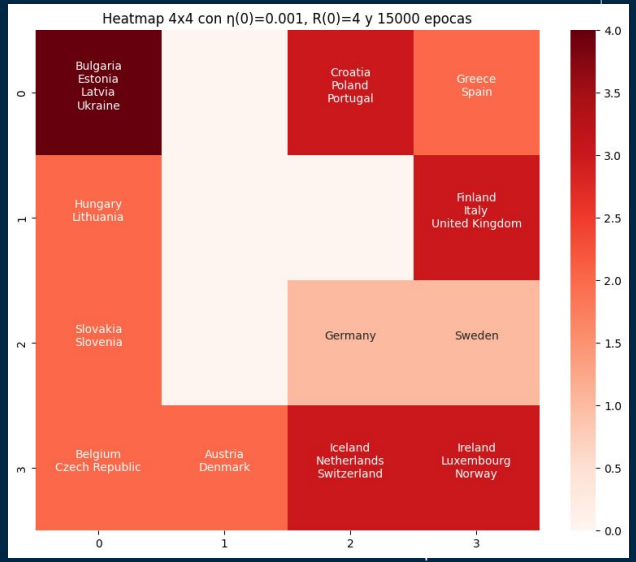


Resultados obtenidos con K=4

Variando la η



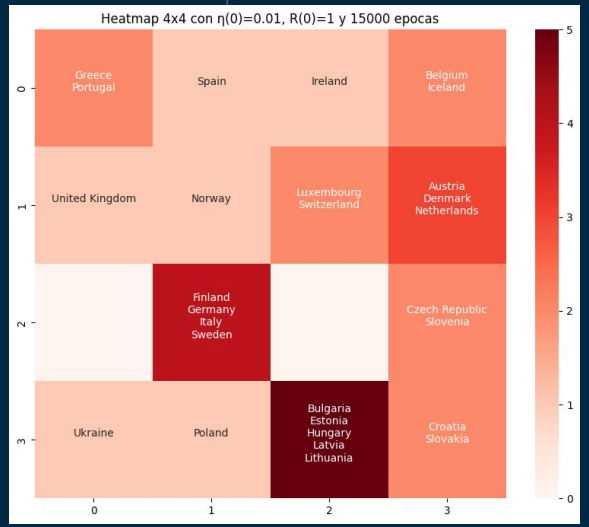


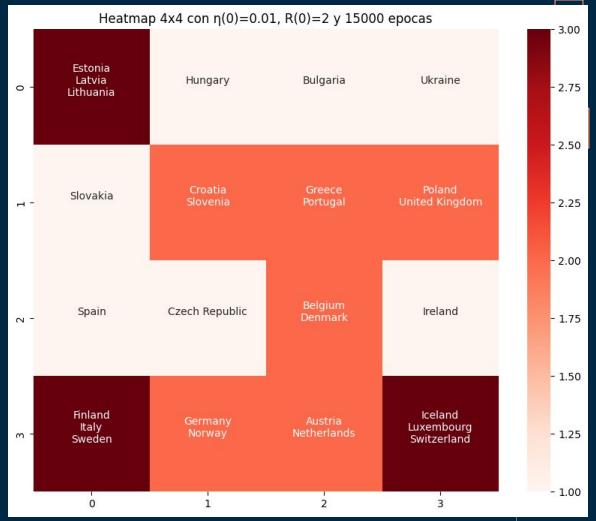


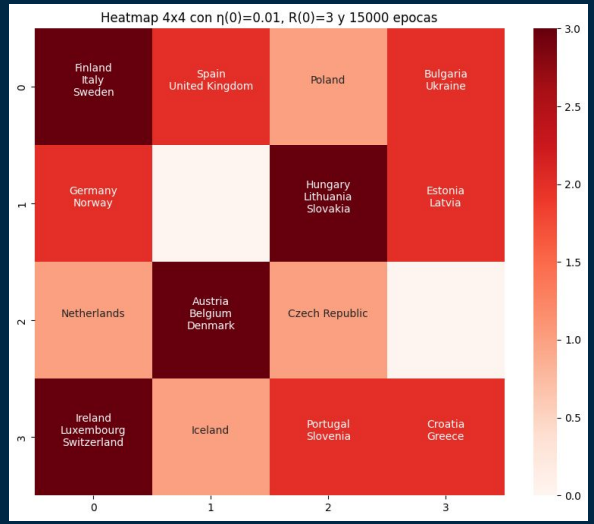
Cuanto menor valor de η, se convergerá a una solución donde los países quedan más agrupados.

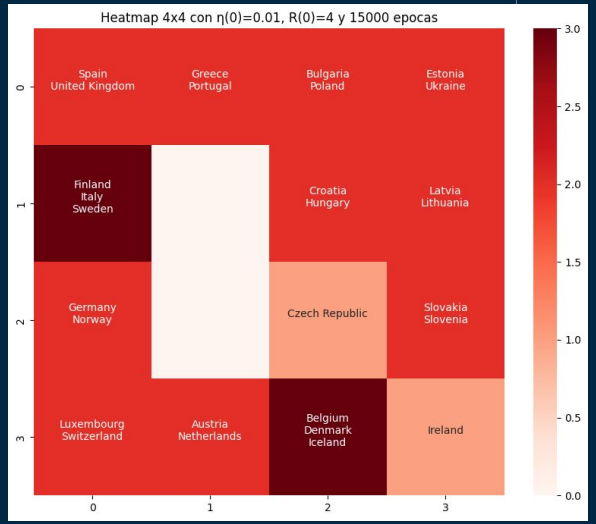
Variando el radio

A mayor radio los cambios se distribuirán más en la red, modificando los pesos en mayor medida en cada iteración.

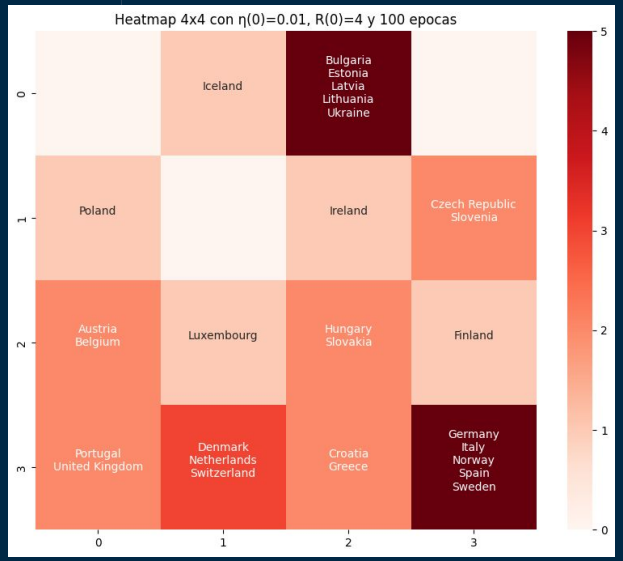


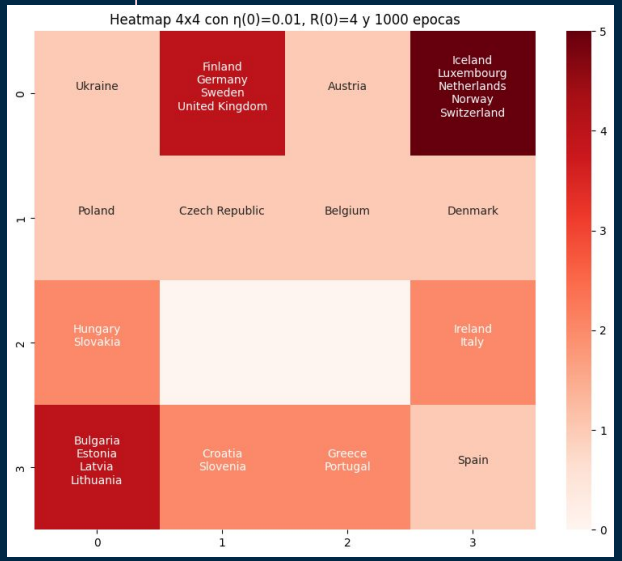


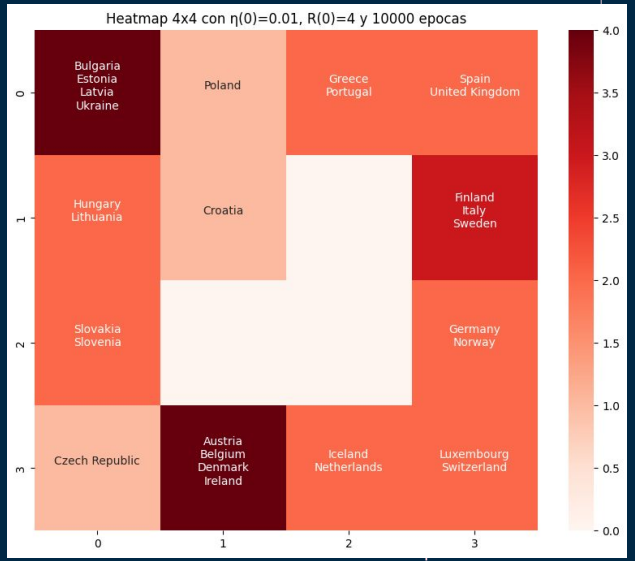




Variando las épocas

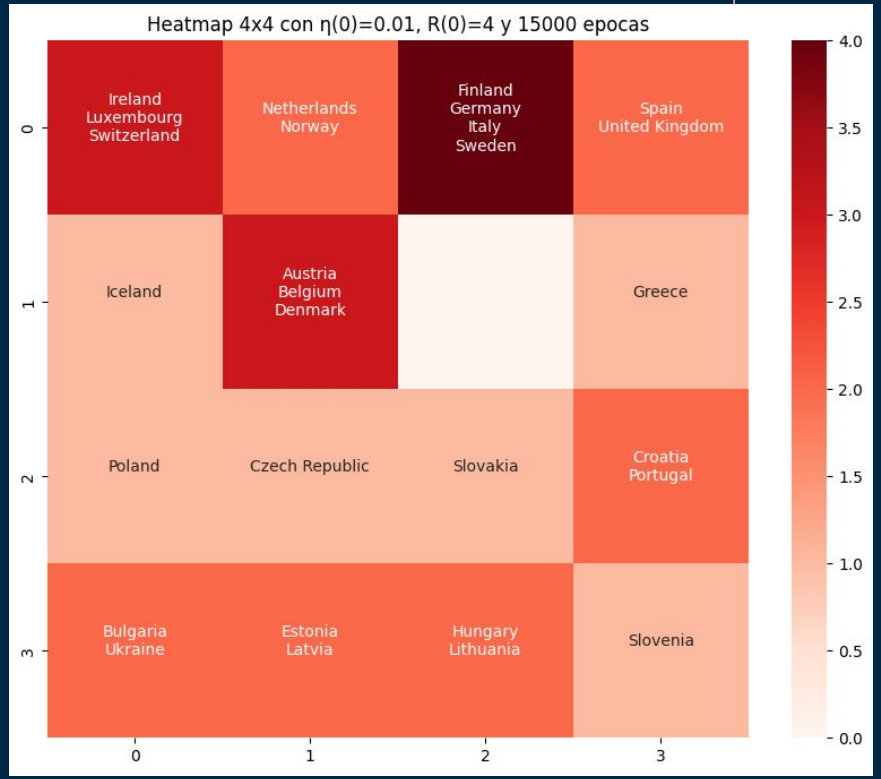


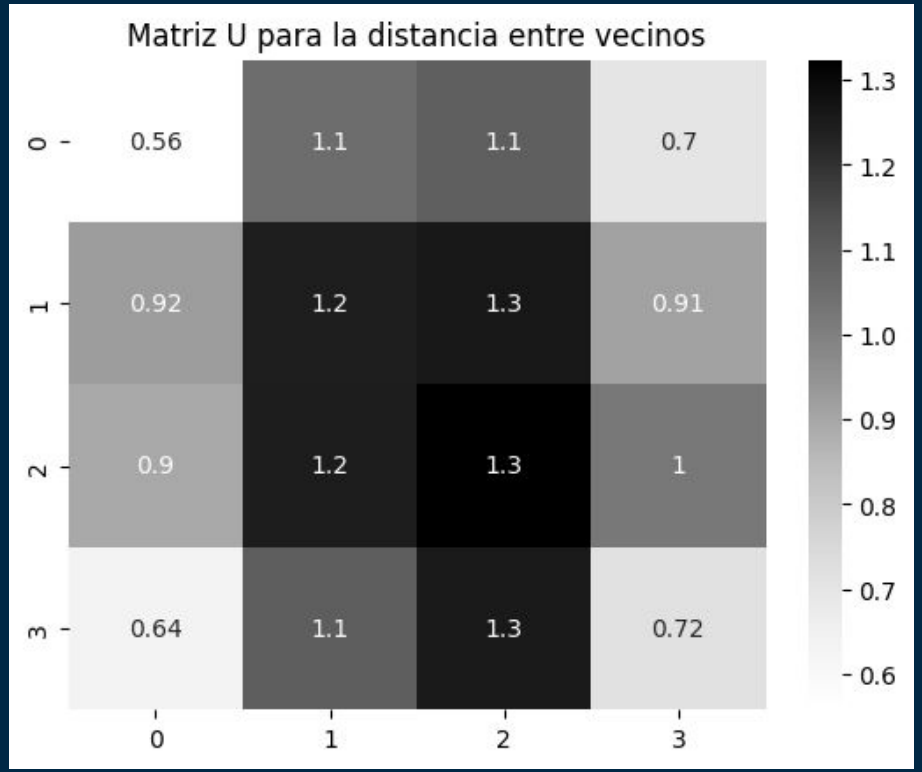




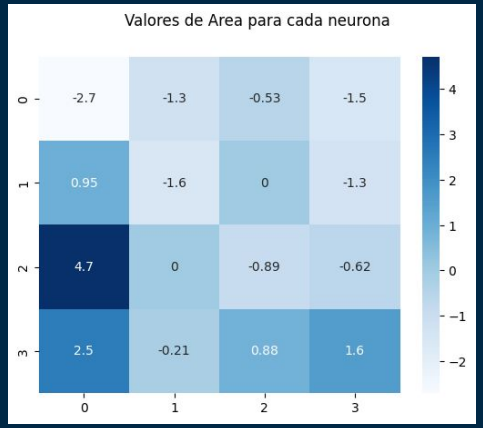
Al tener un valor más grande de cantidad de épocas, los países quedarán más distribuidos, ya que la red tendrá más tiempo para "aprender" como clasificarlos.

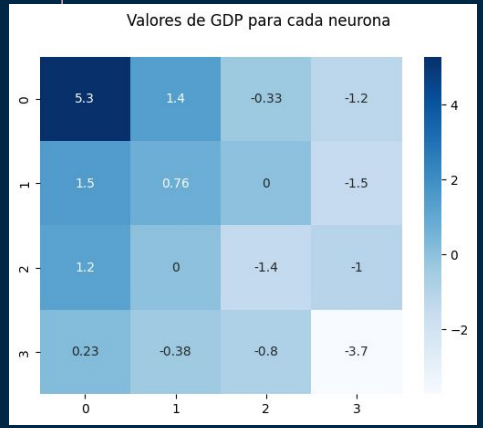
Heatmap y Matriz U

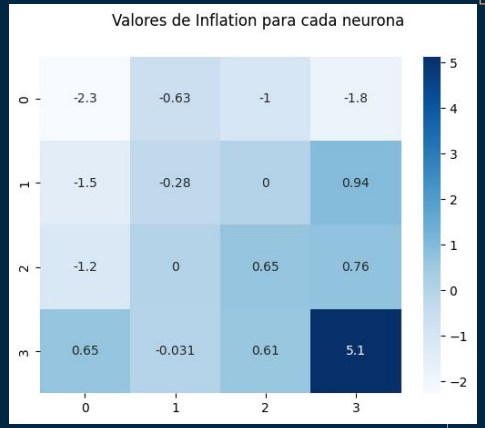


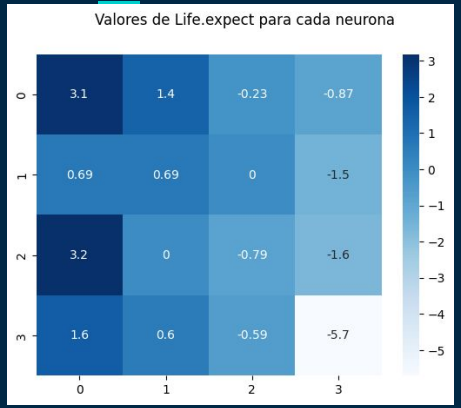


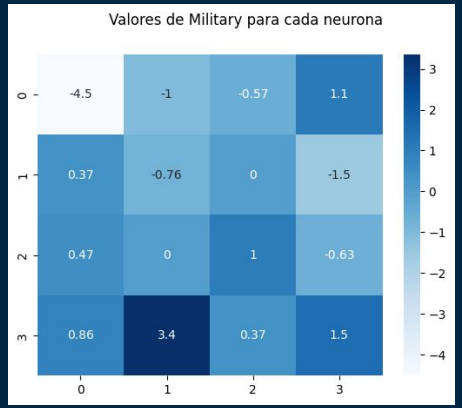
Heatmap de una sola variable

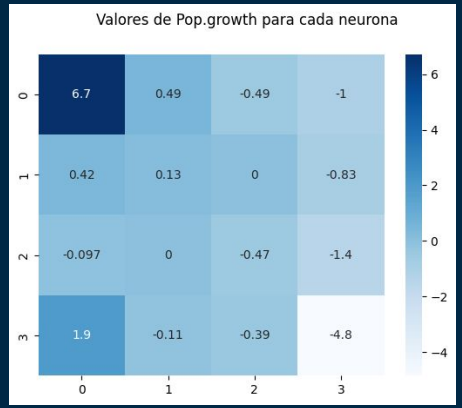


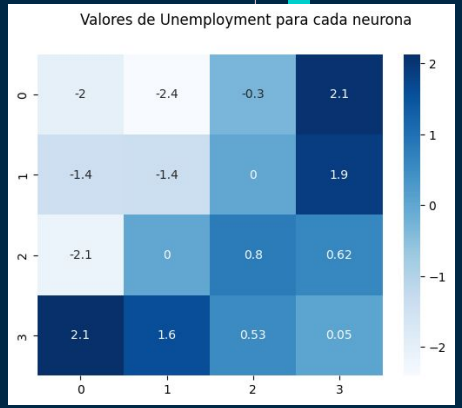












Conclusiones del Problema

- Podemos ver una correlación negativa en los mapas de calor de componentes entre la inflación y desempleo con el GDP. Una mayor inflación o desempleo activan neuronas opuestas a las activadas por el GDP.
- Se puede ver qué expectativa de vida y crecimiento poblacional activan neuronas similares, por lo que podemos decir que ambas tienen una correlación positiva.
- Las variables de área y militarización parecen no tener tanta correlación con las demás, lo cual tiene sentido porque es difícil medir el impacto que estas tienen sobre los otros parámetros mencionados anteriormente.

Regla de Oja

02

Problema

Se cuenta con el mismo conjunto de datos del problema anterior.

Lo que se busca ahora es:

- Calcular la primera componente principal del conjunto de datos utilizando la regla de Oja e interpretar este resultado.
- Comparar el resultado anterior con el que se obtiene si se calcula la primera componente principal con una librería (Sklearn).

Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
Austria	83871	41600	3,5	79,91	0,8	0,03	4,2
Belgium	30528	37800	3,5	79,65	1,3	0,06	7,2
Bulgaria	110879	13800	4,2	73,84	2,6	-0,8	9,6
Croatia	56594	18000	2,3	75,99	2,39	-0,09	17,7
Czech Republic	78867	27100	1,9	77,38	1,15	-0,13	8,5
Denmark	43094	37000	2,8	78,78	1,3	0,24	6,1

PCA (Análisis de componentes principales)

La idea de este método es eliminar la redundancia de las variables cuando están muy correlacionadas entre sí.

Para ello se transforma el conjunto original de variables en uno de nuevas variables que son combinaciones lineales de las anteriores, pero que no están correlacionadas entre sí y tienen la mayor variabilidad posible. A este nuevo conjunto se lo llama ***conjunto de componentes principales*** y se utiliza para extraer características importantes de un conjunto de datos.

$$y_1 = \sum_{j=1}^{p} a_{1j}(x_j - \bar{x_j}) = a_{11}(x_1 - \bar{x_1}) + ... + a_{1p}(x_p - \bar{x_p})$$

Regla de Oja

Existen algunos modelos de redes neuronales que permiten calcular las componentes principales en forma iterativa. Una forma es utilizando un perceptrón simple lineal de aprendizaje Hebbiano:

$$O^{\mu} = \theta\left(\sum_{i=1}^{n} x_i^{\mu} w_i^{\mu}\right) \qquad \Delta w = \eta O^{\mu} x^{\mu}$$

Oja demostró que si la red anterior convergiese, el vector de pesos resultante converge al autovector correspondiente al mayor autovalor de la matriz de correlaciones de los datos de entrada (es decir, la primera componente principal).

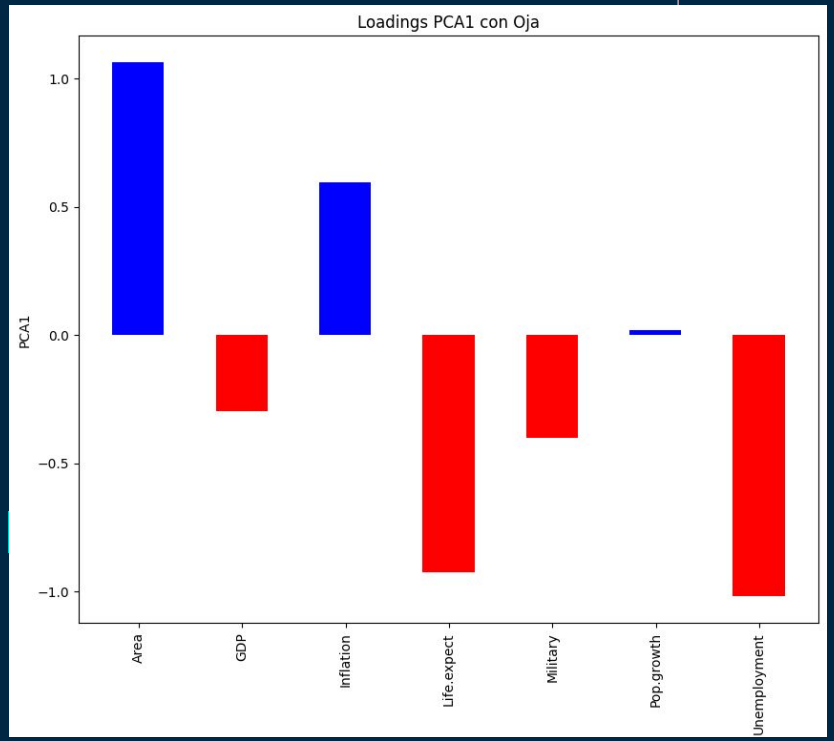
$$y_1 = a_1 x_1 + a_2 x_2 + \ldots$$

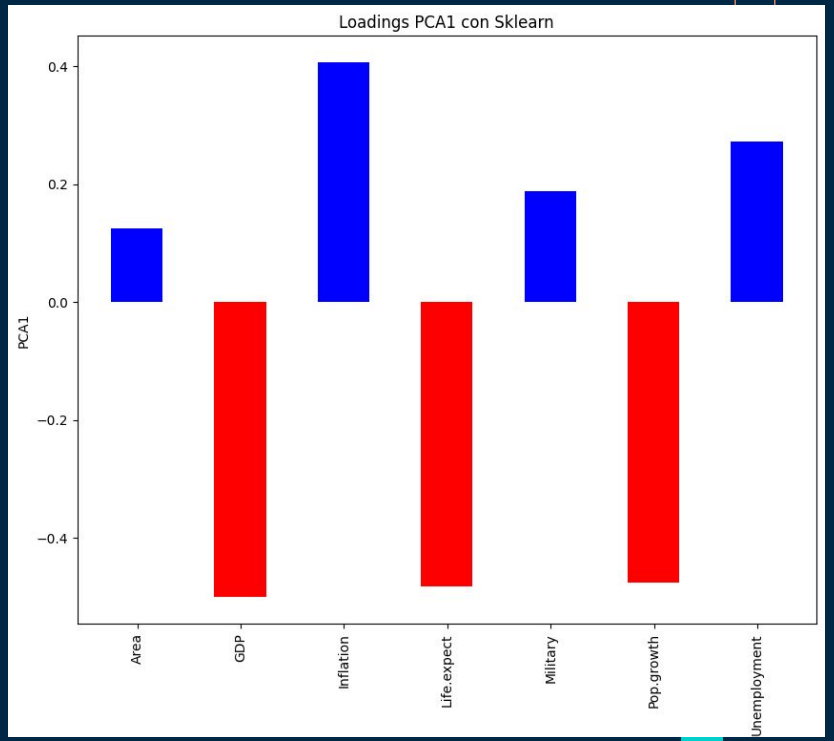
Para lograrlo, se deben actualizar los pesos con la siguiente regla:

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

Resultados obtenidos

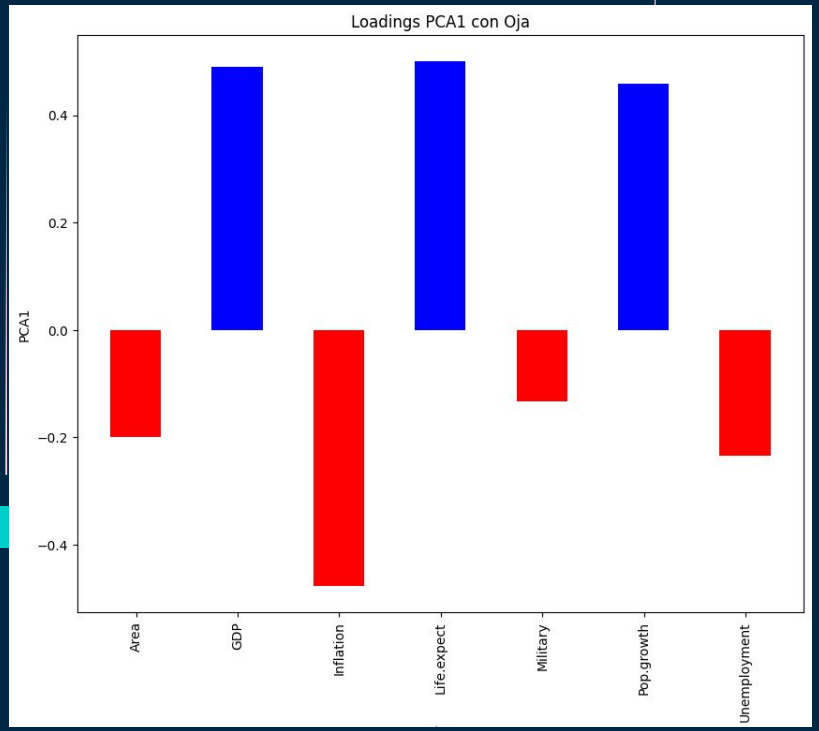
η=0.1 y 100.000 épocas

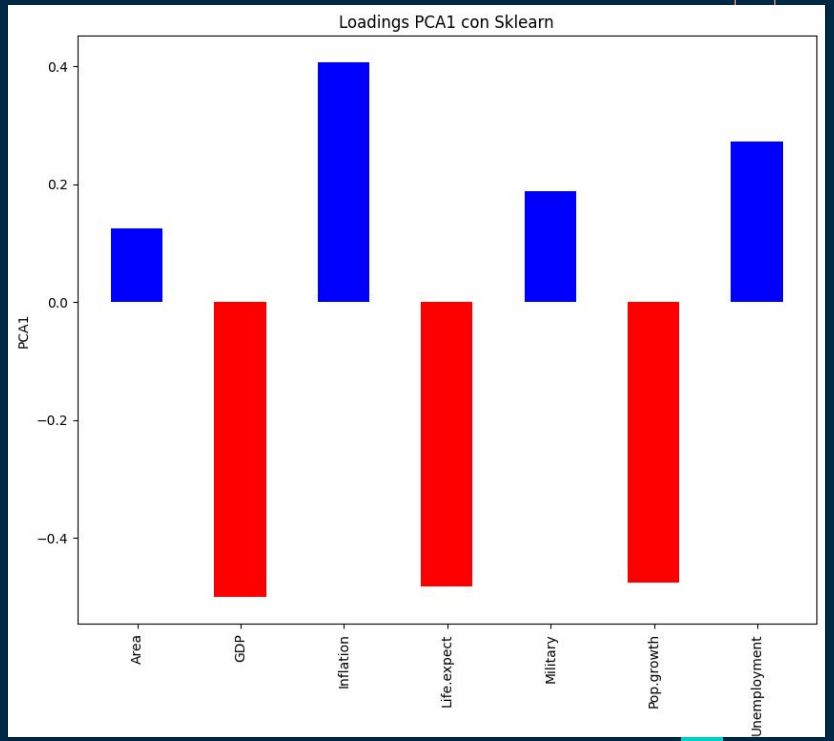




	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
OJA	1.0625	-0.2975	0.5952	-0.9241	-0.4004	0.0195	-1.0179
SKLEARN	0.1249	-0.5005	0.4065	-0.4829	0.1881	-0.4757	0.2717

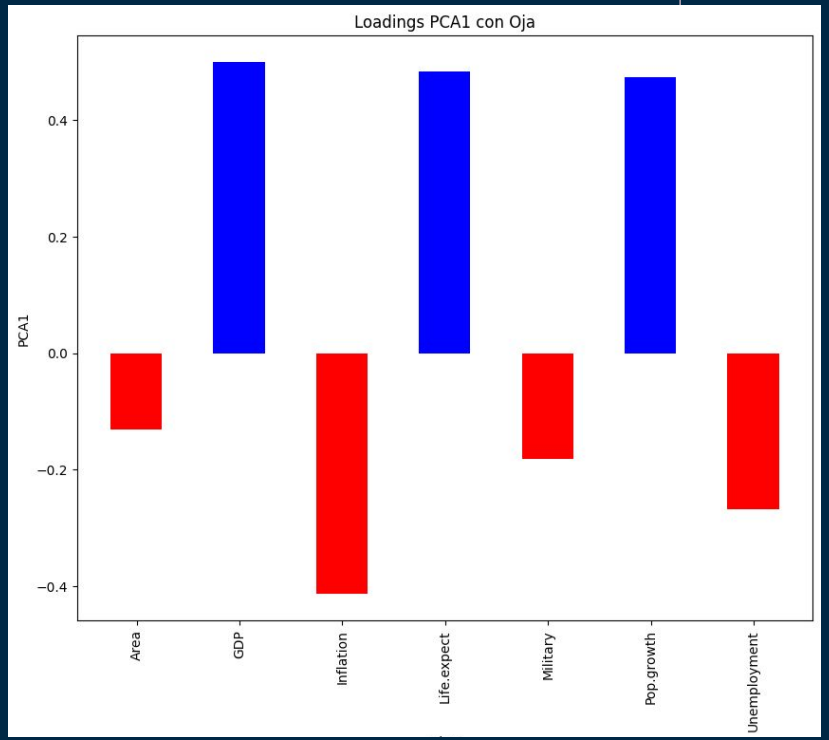
η=0.01 y 100.000 épocas

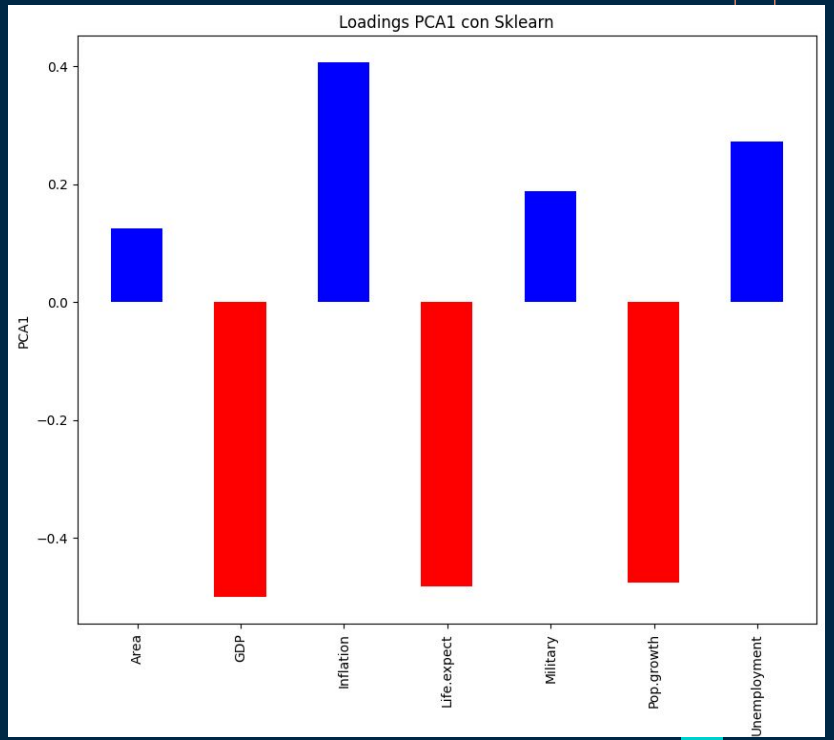




	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
OJA	-0.1998	0.4905	-0.4768	0.5002	-0.1327	0.4576	-0.2336
SKLEARN	0.1249	-0.5005	0.4065	-0.4829	0.1881	-0.4757	0.2717

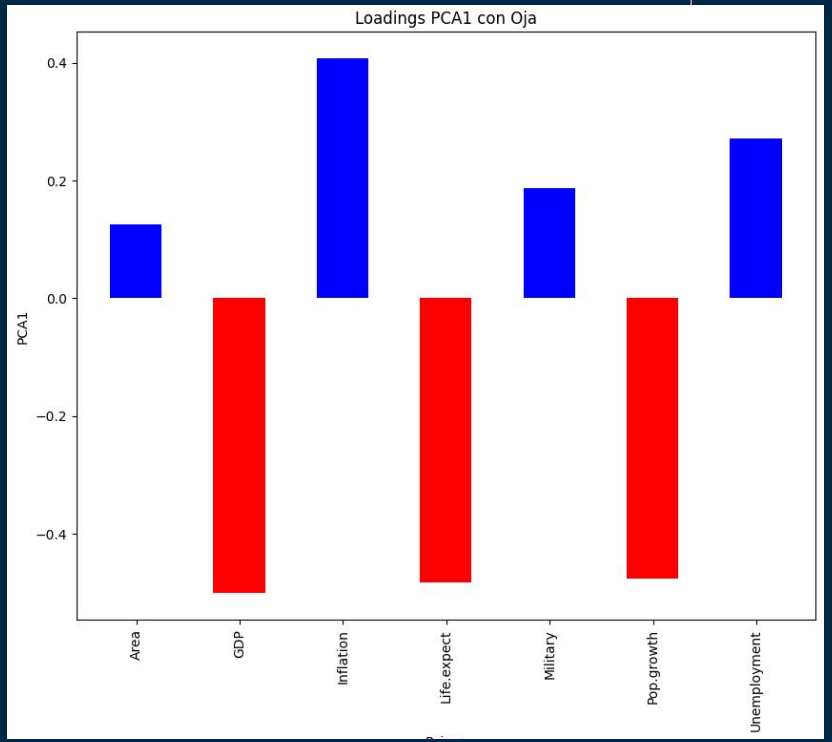
η=0.001 y 100.000 épocas

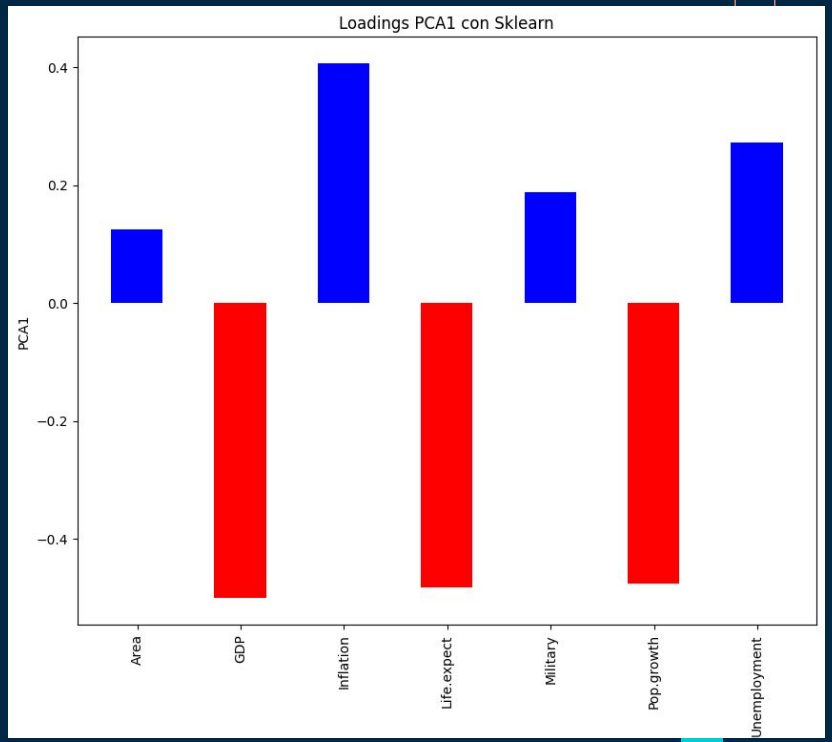




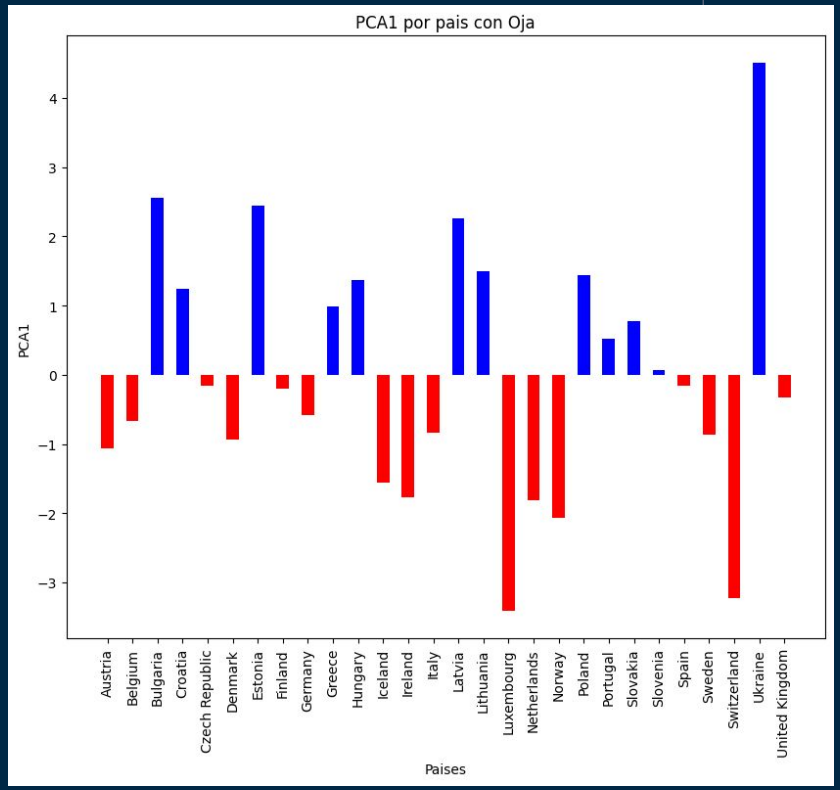
	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
OJA	-0.1318	0.4999	-0.4133	0.4843	-0.1824	0.4742	-0.2683
SKLEARN	0.1249	-0.5005	0.4065	-0.4829	0.1881	-0.4757	0.2717

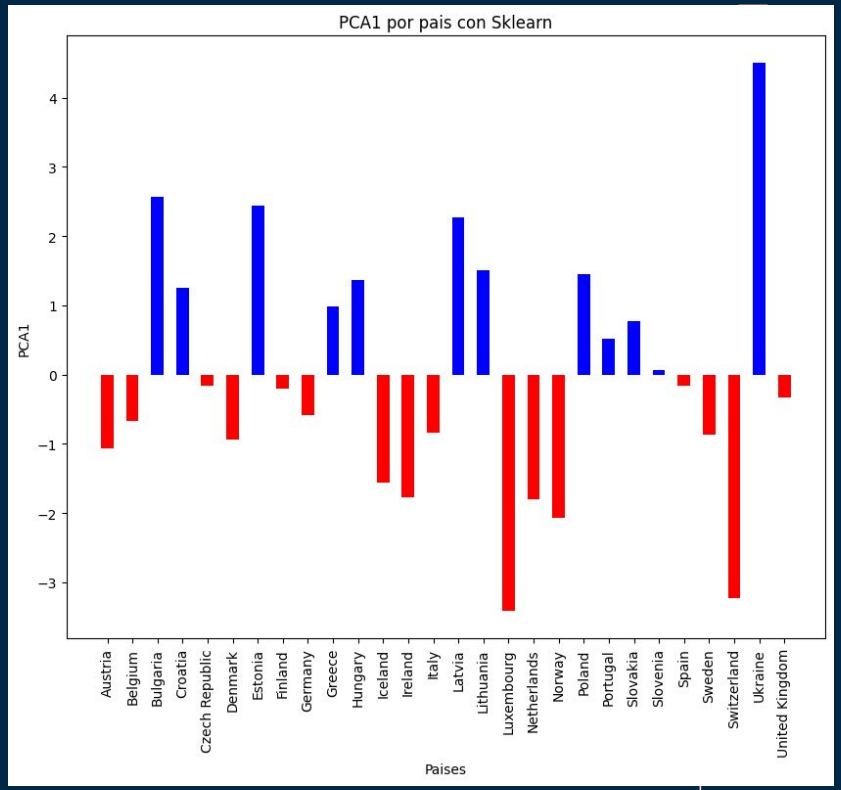
η=0.0001 y 100.000 épocas





	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
OJA	0.1256	-0.5004	0.4072	-0.4830	0.1875	-0.4756	0.2713
SKLEARN	0.1249	-0.5005	0.4065	-0.4829	0.1881	-0.4757	0.2717



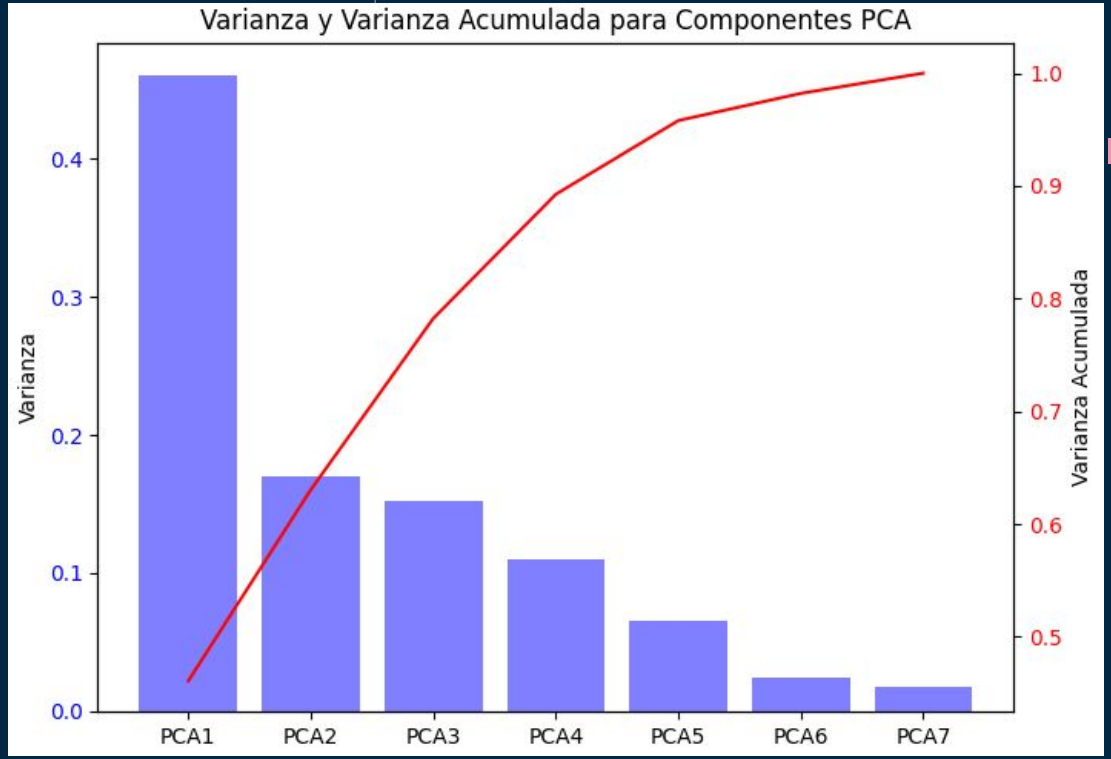


	Austria	Belgium	Bulgaria	Croatia	Czech Republic	Denmark	Estonia	Finland	Germany	Greece	Hungary	Iceland	Ireland	Italy	Latvia	Lithuania	Luxembourg
OJA	-1.0616	-0.6690	2.5622	1.2452	-0.1649	-0.9382	2.4427	-0.2063	-0.5811	0.9797	1.3715	-1.5537	-1.7767	-0.8376	2.2647	1.5025	-3.4152
SKLEARN	-1.0623	-0.6688	2.5629	1.2473	-0.1642	-0.9380	2.4429	-0.2068	-0.5817	0.9824	1.3717	-1.5552	-1.7763	-0.8378	2.2645	1.5025	-3.4158

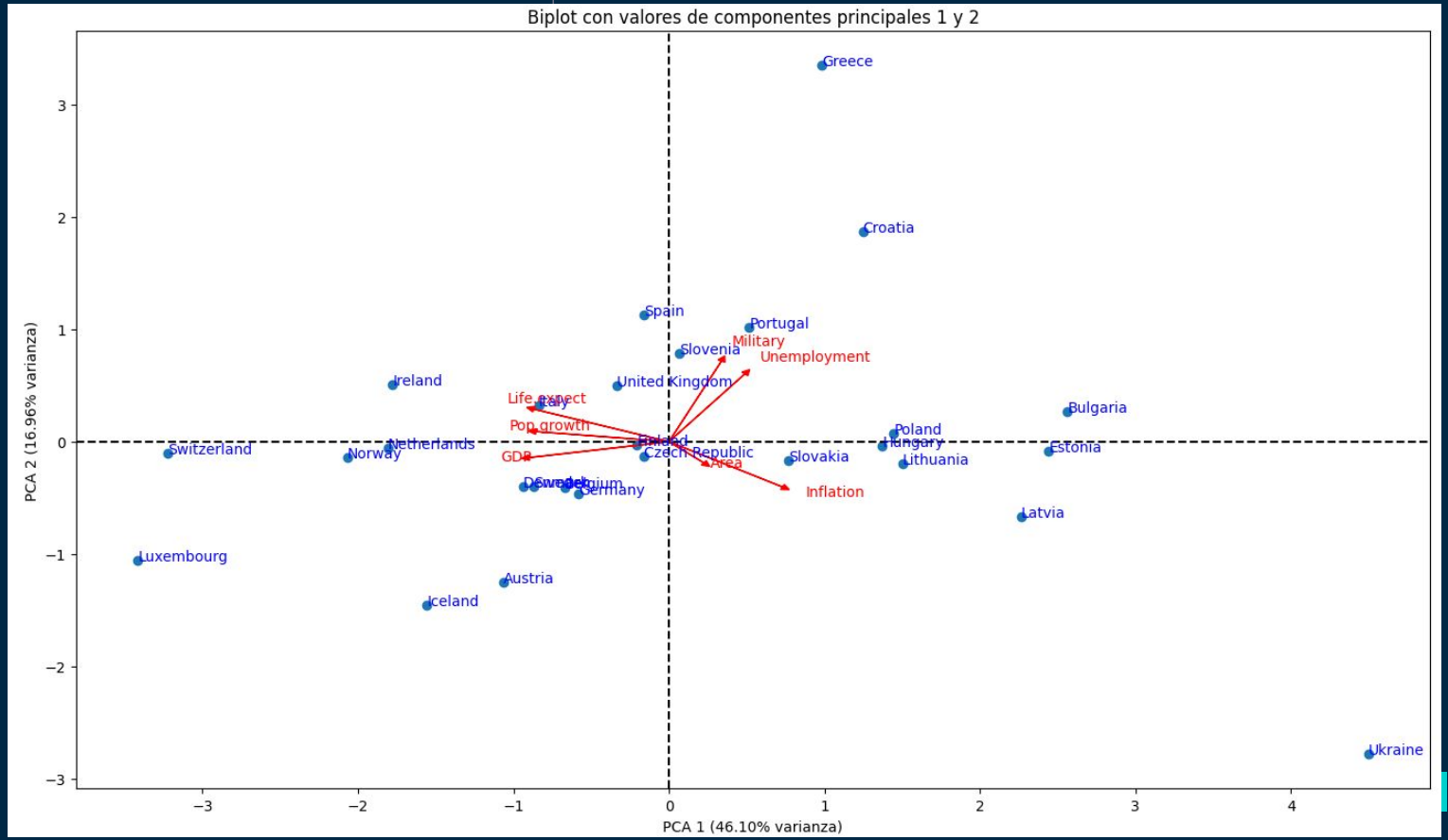
Netherlands	Norway	Poland	Portugal	Slovakia	Slovenia	Spain	Sweden	Switzerland	Ukraine	United Kingdom
-1.8075	-2.0686	1.4459	0.5161	0.7688	0.0647	-0.1601	-0.8680	-3.2234	4.5022	-0.3344
-1.8069	-2.0686	1.4453	0.5170	0.7689	0.0663	-0.1608	-0.8692	-3.2225	4.4977	-0.3347

η=0.0001 y 100.000 épocas

Análisis extra de PCA con Sklearn



	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7
Varianza	0.4610	0.1696	0.1519	0.1101	0.0654	0.0241	0.0179
Varianza acumulada	0.4610	0.6306	0.7825	0.8925	0.9580	0.9821	1.0000



La variable con menor influencia es el área, mientras que la inflación es la más característica.

Conclusiones de Oja

- La red neuronal con regla de Oja logra aproximar correctamente la primera componente principal con un valor de η adecuado.
- A menor tasa de aprendizaje, se obtienen mejores resultados.
- Con mayor cantidad de épocas, se logra una mejor convergencia a los resultados.
- Si PCA1 > 0, el país se ve más afectado por área, inflación, desempleo y nivel militar.
- Si PCA1 < 0, el país se ve más afectado por PBI, crecimiento poblacional y expectativa de vida.
- Puede ocurrir que Oja converja al autovector opuesto comparado con el obtenido con la librería Sklearn.

Modelo de Hopfield

03

Red de Hopfield

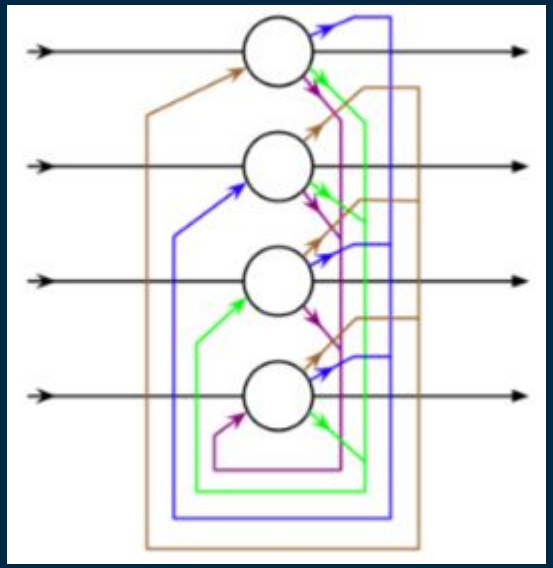
Una red de Hopfield es una red neuronal donde todas las neuronas están conectadas entre sí (pero no consigo mismas) que sirve para asociar un patrón de consulta binario (con perturbaciones o ruido) con alguno de los patrones almacenados utilizando el concepto de memoria asociativa (recuperación de información por asociación con datos almacenados).

Cada neurona tiene 2 posibles estados:

$S_i = +1$ (activo) $S_i = -1$ (inactivo)

Limitaciones:

- El número máximo de patrones que puede almacenar es igual al 15% del número de neuronas de la red.
- Los patrones deben ser "más o menos" ortogonales.



Funcionamiento de Hopfield

1. Almaceno los $\xi_1, \xi_2, \ldots \xi_p$ patrones N-dimensionales binarios
2. Calculo los pesos sinápticos

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases}$$

3. Inicializar $S_i(0) = \zeta_i$, donde ζ_i es el patrón de consulta.
4. Iterar y actualizar los elementos del vector de estado S(t) hasta alcanzar la convergencia de acuerdo de la regla:

$$S_i(t+1) = sign\left(\sum_{j=1}^{N} w_{ij} S_j(t)\right), \quad \text{con } i \neq j$$

La convergencia ocurre cuando se alcanza un estado estable (es decir, el estado calculado no se modificó con respecto al estado previo).

Energía y estados espúreos

Hopfield demostró que se puede asociar una función de energía a la red, donde los mínimos locales de dicha función representan los patrones almacenados:

$$H(w) = -\frac{1}{2}\sum_{i,j} w_{ij} S_i S_j$$

Una propiedad central de la función de energía es que siempre decrece (o permanece constante) cuando el sistema evoluciona.

Puede ocurrir que esta función de energía también puede tener otros mínimos locales que no son los patrones almacenados, y, por lo tanto, no siempre se va a poder asociar con un patrón almacenado y llegar a un estado estable.

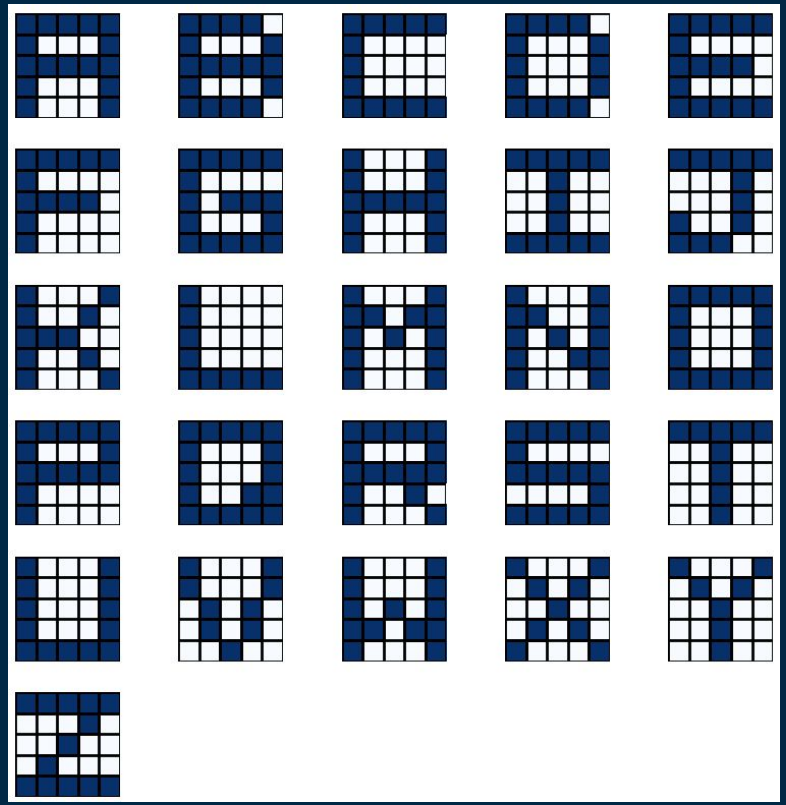
Estados espúreos

Problema

Se construyeron patrones de letras en matrices de 5x5 representadas con 1 y –1.

La idea es implementar una red de Hopfield en la que se almacenan 4 patrones y se busca asociar matrices ruidosas de 5x5 con alguno de los patrones almacenados.

¿Por qué se almacenan solo 4 patrones? Se recomienda que una red de Hopfield solo almacene como máximo un 15% del tamaño del patrón, entonces 15% de 25 ≈ 4



Ortogonalidad de las letras

Ninguna combinación de los posibles patrones a almacenar será 100% ortogonal, ya que la cantidad de elementos de cada patrón es impar (25, puesto que es una matriz de 5x5).

Por lo tanto, se realiza el siguiente análisis, teniendo en cuenta todas las posibles permutaciones de letras, para comprobar qué conjuntos son mejores para almacenar en la red:

```
('A', 'B', 'C', 'D')
[[ 0 15  7  9]
 [15  0  9 19]
 [ 7  9  0 15]
 [ 9 19 15  0]]
```

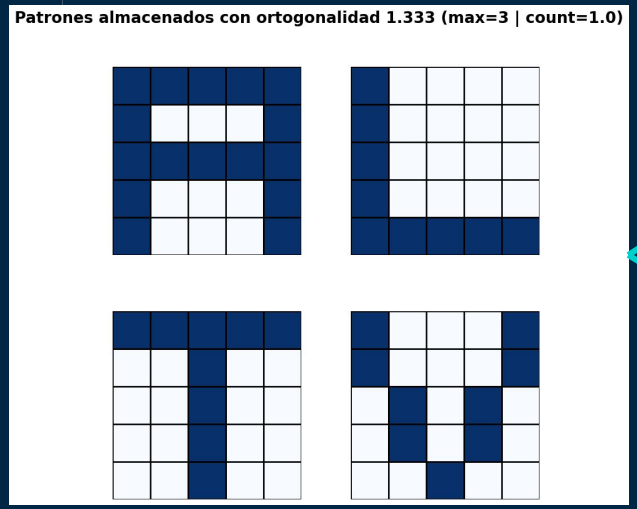
Medio: representa la media de ortogonalidad entre el conjunto de letras.
Max: representa el valor máximo de la matriz de ortogonalidades entre las letras.
Count: representa cuántas veces aparece el valor "max".

|<,>| max	|<,>| medio	grupo
max: 3 | count: 1	1.33	('A', 'L', 'T', 'V')
max: 3 | count: 1	1.33	('L', 'R', 'T', 'X')
max: 3 | count: 2	1.67	('A', 'J', 'L', 'V')
max: 3 | count: 2	1.67	('F', 'I', 'U', 'X')
max: 3 | count: 2	1.67	('I', 'R', 'U', 'X')
max: 3 | count: 2	1.67	('L', 'R', 'T', 'V')
max: 3 | count: 3	2.00	('F', 'U', 'V', 'Z')
max: 3 | count: 3	2.00	('L', 'P', 'T', 'V')
max: 3 | count: 3	2.00	('P', 'U', 'V', 'Z')
max: 3 | count: 3	2.00	('R', 'U', 'V', 'Z')

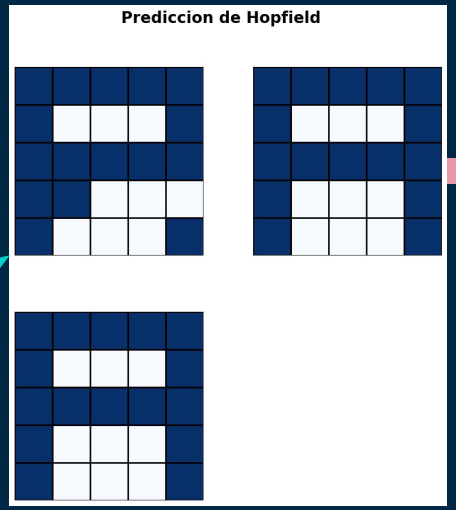
|<,>| max	|<,>| medio	grupo
max: 23 | count: 1	8.67	('O', 'Q', 'V', 'W')
max: 23 | count: 1	8.67	('O', 'Q', 'V', 'X')
max: 23 | count: 1	9.33	('O', 'Q', 'V', 'Y')
max: 23 | count: 1	7.67	('O', 'Q', 'V', 'Z')
max: 23 | count: 1	10.67	('O', 'Q', 'W', 'X')
max: 23 | count: 1	10.33	('O', 'Q', 'W', 'Y')
max: 23 | count: 1	9.00	('O', 'Q', 'W', 'Z')
max: 23 | count: 1	12.00	('O', 'Q', 'X', 'Y')
max: 23 | count: 1	10.00	('O', 'Q', 'X', 'Z')
max: 23 | count: 1	10.00	('O', 'Q', 'Y', 'Z')

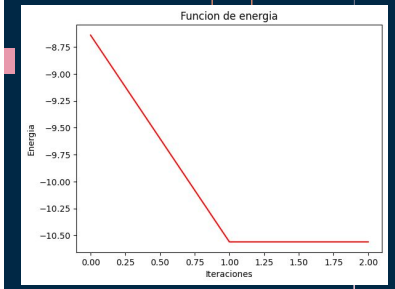
Podemos observar que las letras más parecidas son menos ortogonales.

Resultados obtenidos

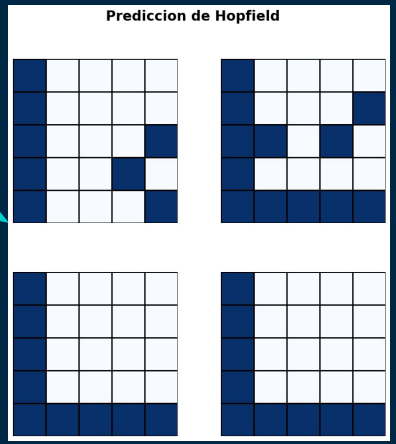


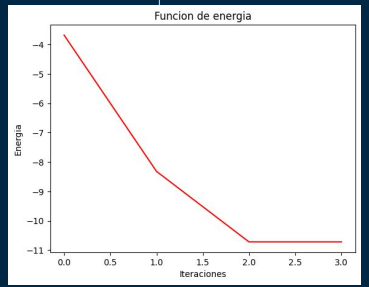
Patrón de consulta
A mutada con prob 0.1

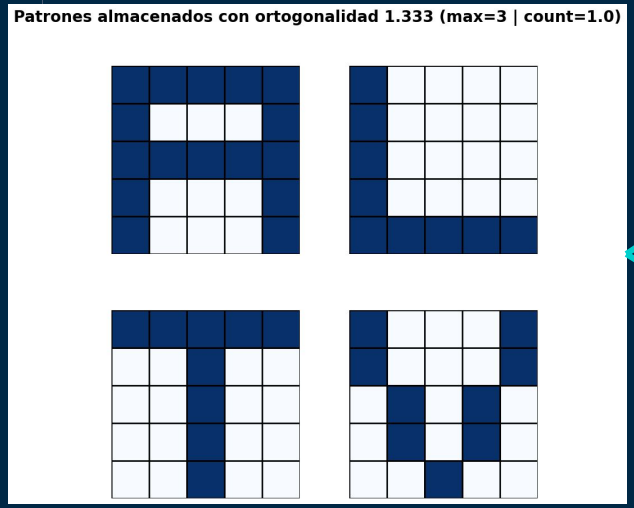




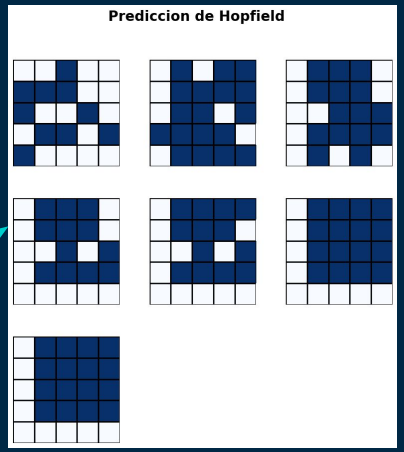
Patrón de consulta
L mutada con prob 0.2

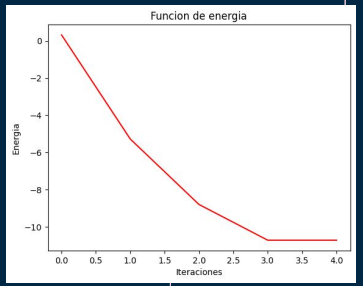




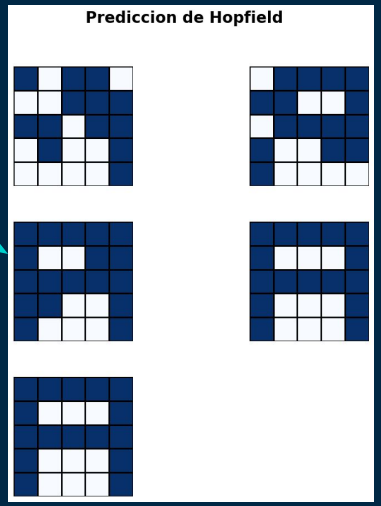


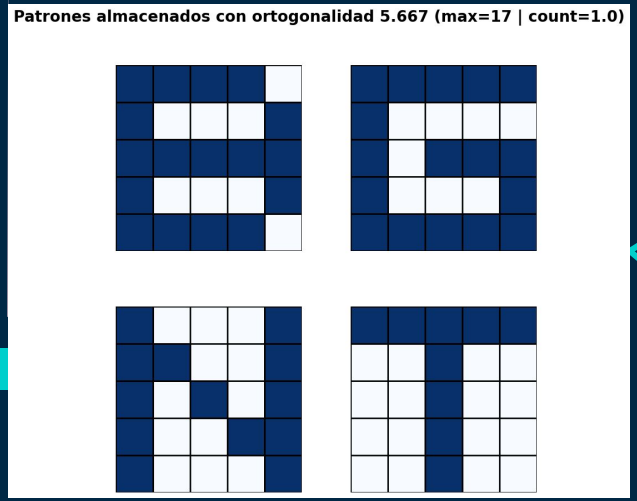
Patrón de consulta
T mutada con prob 0.5



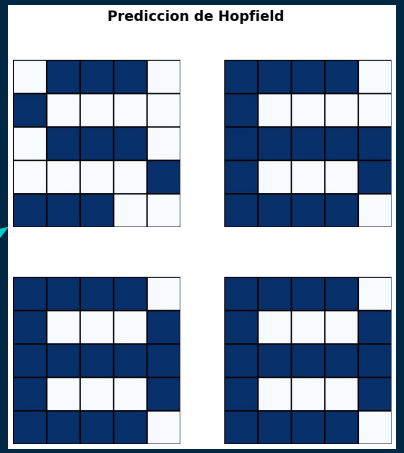


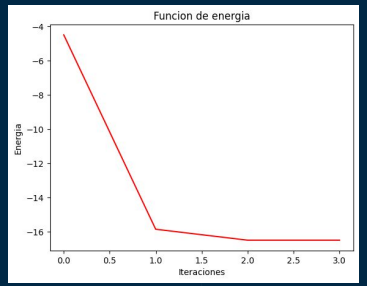
Patrón de consulta
A mutada con prob 0.5



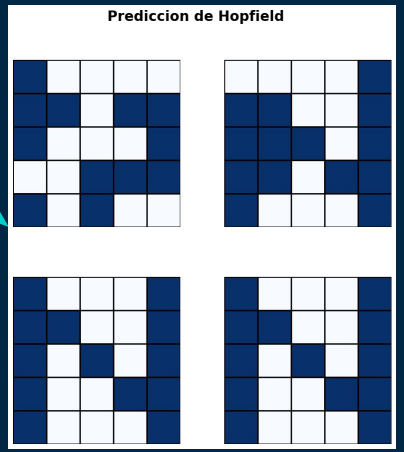


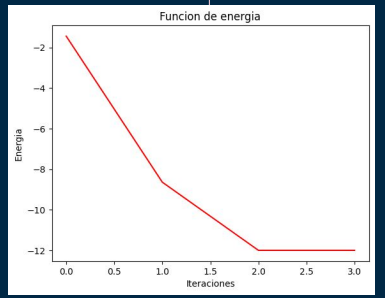
Patrón de consulta
B mutada con prob 0.2



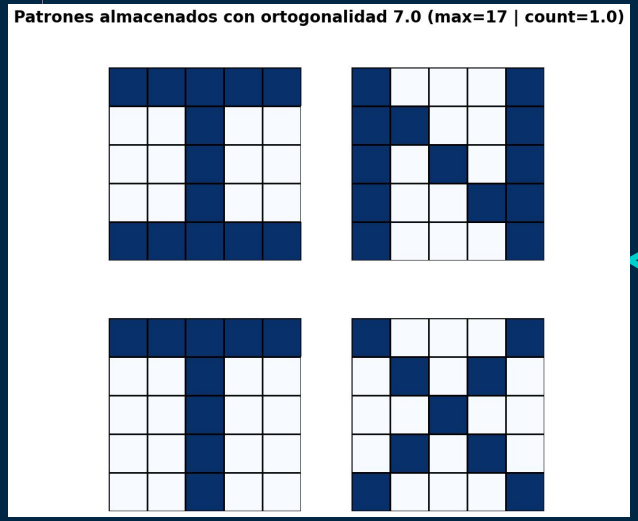


Patrón de consulta
N mutada con prob 0.2

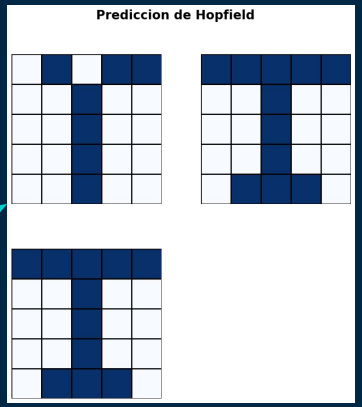


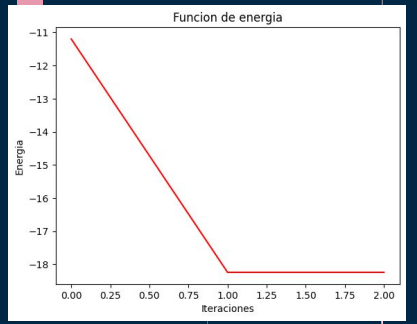


Estados espúreos

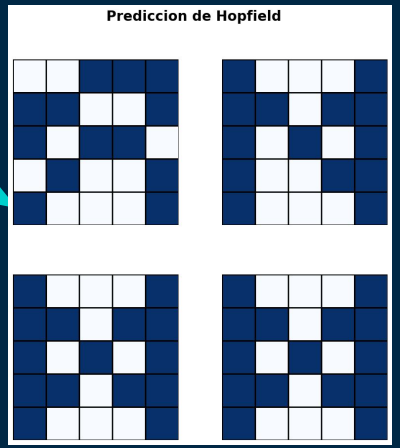


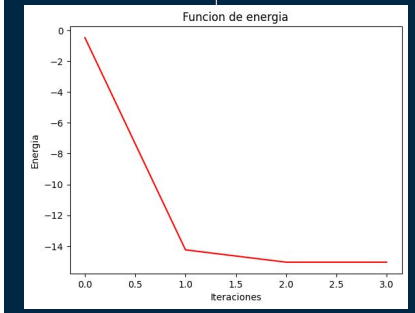
Patrón de consulta
T mutada con prob 0.2



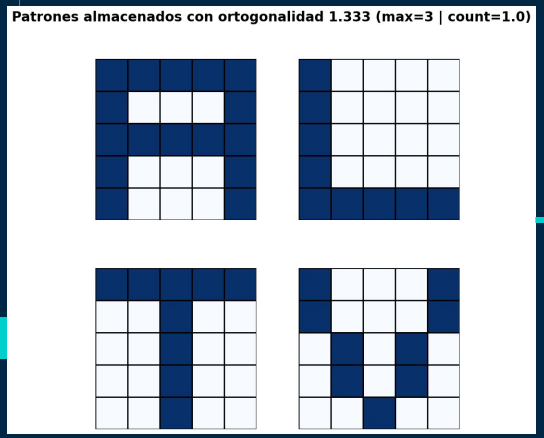


Patrón de consulta
X mutada con prob 0.3

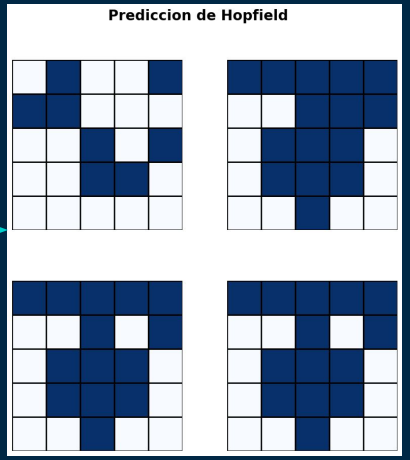


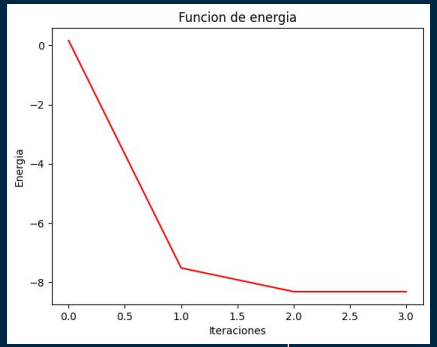


Estados espúreos



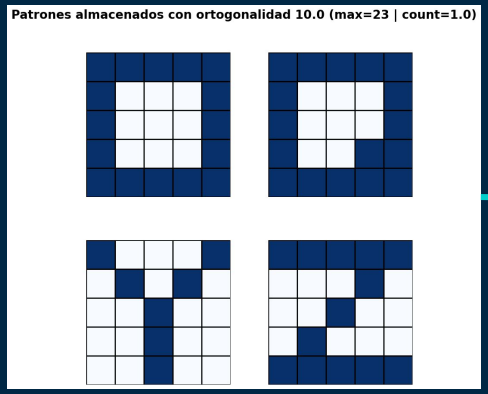
Patrón de consulta
L mutada con prob 0.5



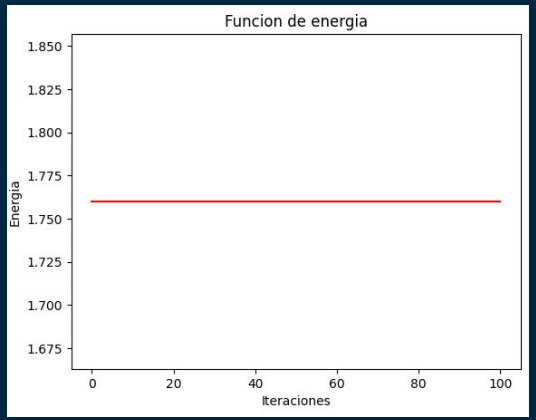


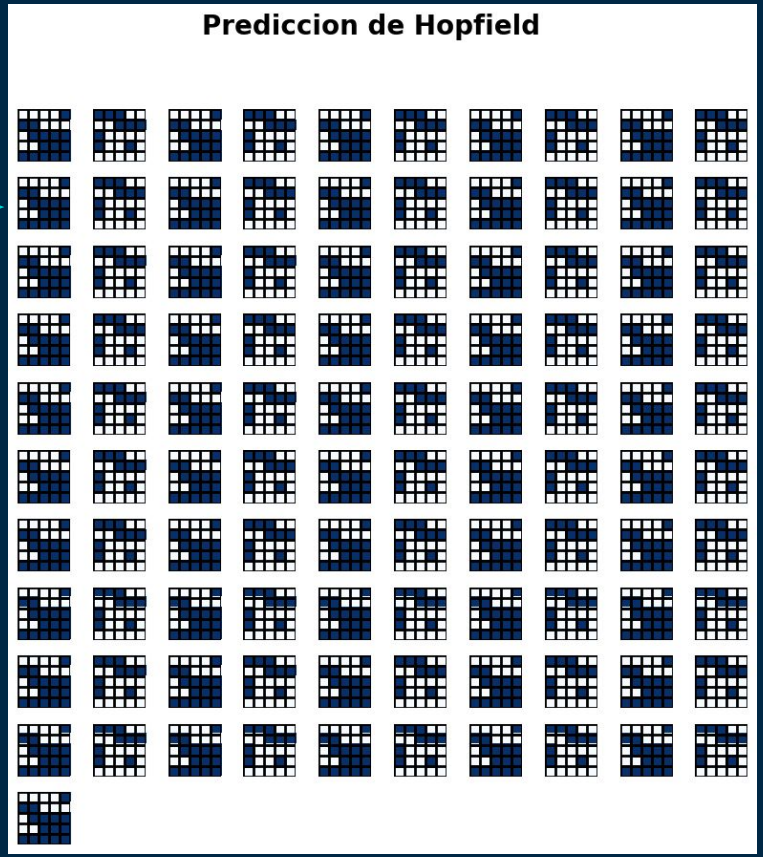
Incluso con el mejor conjunto de letras para almacenar teniendo en cuenta la ortogonalidad, podemos converger en un estado espúreo

Estado espúreo cíclico



Patrón de consulta
O mutada con prob 0.5





Conclusiones de Hopfield

- Cuanto más ortogonales sean los patrones, mejor asocia patrones ruidosos.
- Puede ocurrir que se llegue al patrón correcto, pero de forma “opuesta”.
- A medida que disminuye la ortogonalidad de los patrones almacenados, es más probable que converja a un estado espúreo.
- Con alta probabilidad de ruido, existe la posibilidad de caer en estados espureos, aun con patrones muy ortogonales.
- La función de energía asociada siempre decrece o permanece constante.

¡Gracias!