



SIA - Trabajo práctico 4

Aprendizaje No Supervisado

Grupo 4
Tomás Dallas
Tomás Dorado



Implementación

- Implementación en Python
- Se utilizaron las librerías:
 - **numpy:** para manejo de arrays, generación de randoms y multiplicaciones vectoriales
 - **matplotlib:** para graficar
 - **pandas:** para parseo
 - **skicit-learn:** para PCA
 - **scipy:** para el cálculo de la distancia euclidiana en Kohonen
 - **scikit-image:** para graficar Hopfield
 - **seaborn:** para graficar el heatmap de matriz de correlación



Ejercicio 1

El conjunto de datos europe.csv corresponde a características económicas, sociales y geográficas de 28 países de Europa. Las variables son:

- **Country:** Nombre del país.
- **Area:** Área.
- **GDP:** Producto bruto interno.
- **Inflation:** Inflación anual.
- **Life.expect:** Expectativa de vida media en años.
- **Military:**
- **Pop.growth:** Tasa de crecimiento poblacional.
- **Unemployment:** Tasa de desempleo.

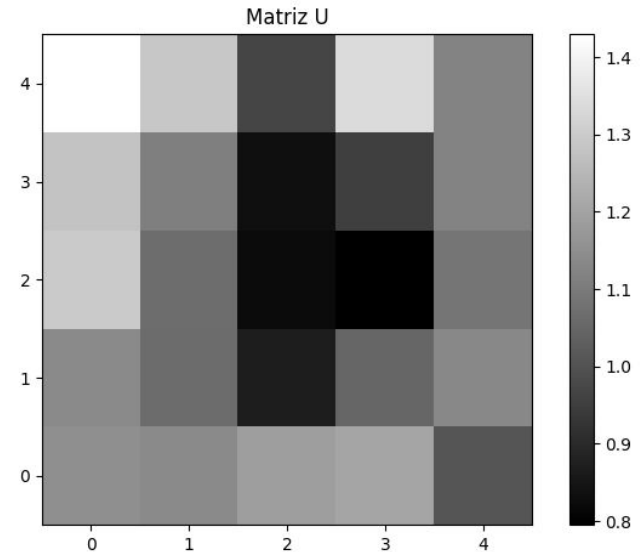
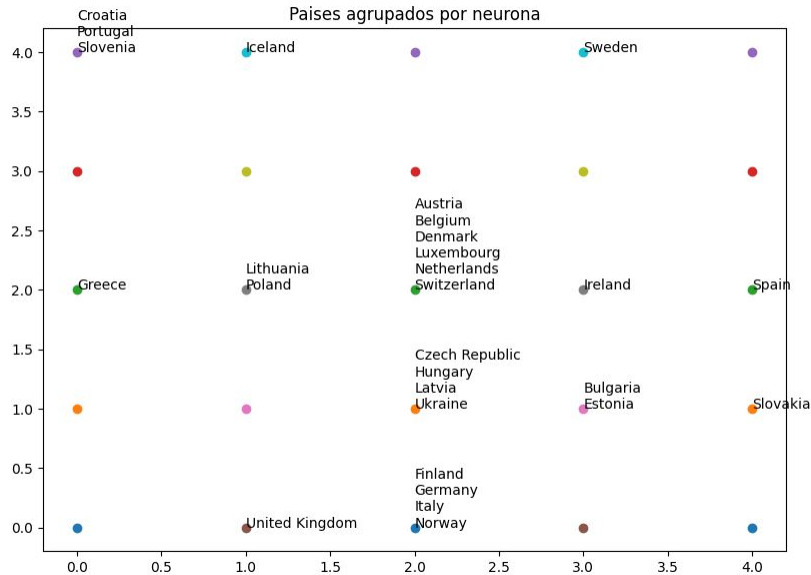


Ejercicio 1

- a) Implementar la red de Kohonen y aplicarla para resolver los siguientes problemas:
 - 1) Asociar países que posean las mismas características geopolíticas, económicas y sociales.
 - 2) Realizar al menos un gráfico que muestre los resultados.
 - 3) Realizar un gráfico que muestre las distancias promedio entre neuronas vecinas.
 - 4) Analizar la cantidad de elementos que fueron asociados a cada neurona.

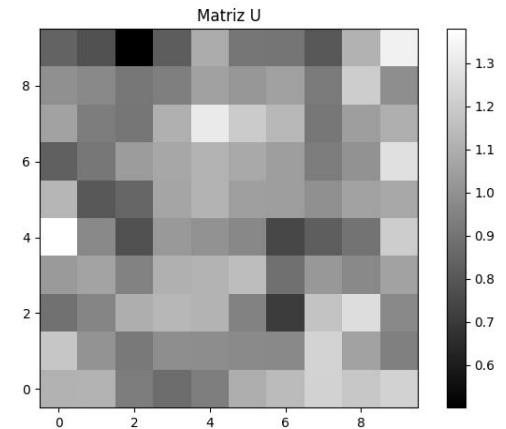
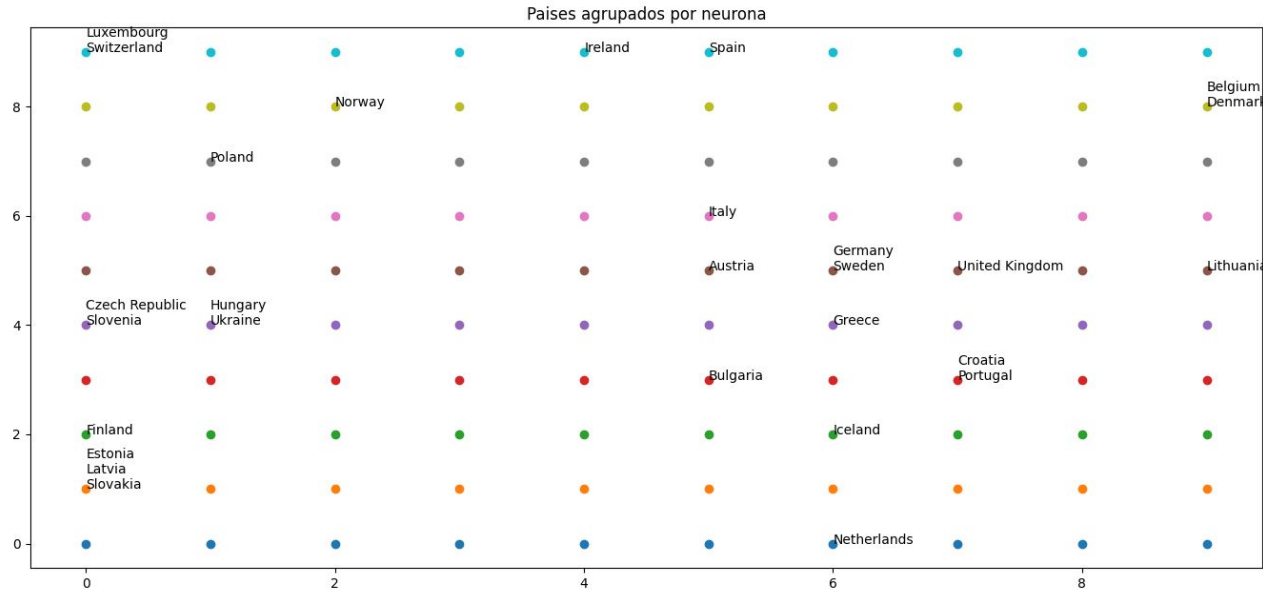
Ejercicio 1a - Resultados

5x5 neuronas, 2500 iteraciones, eta= 0.0001



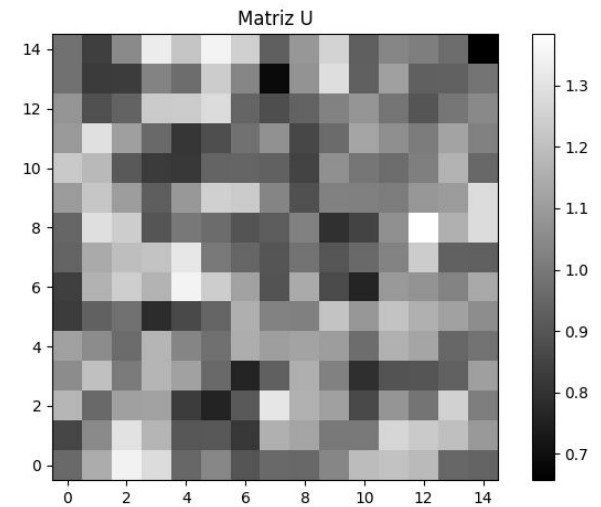
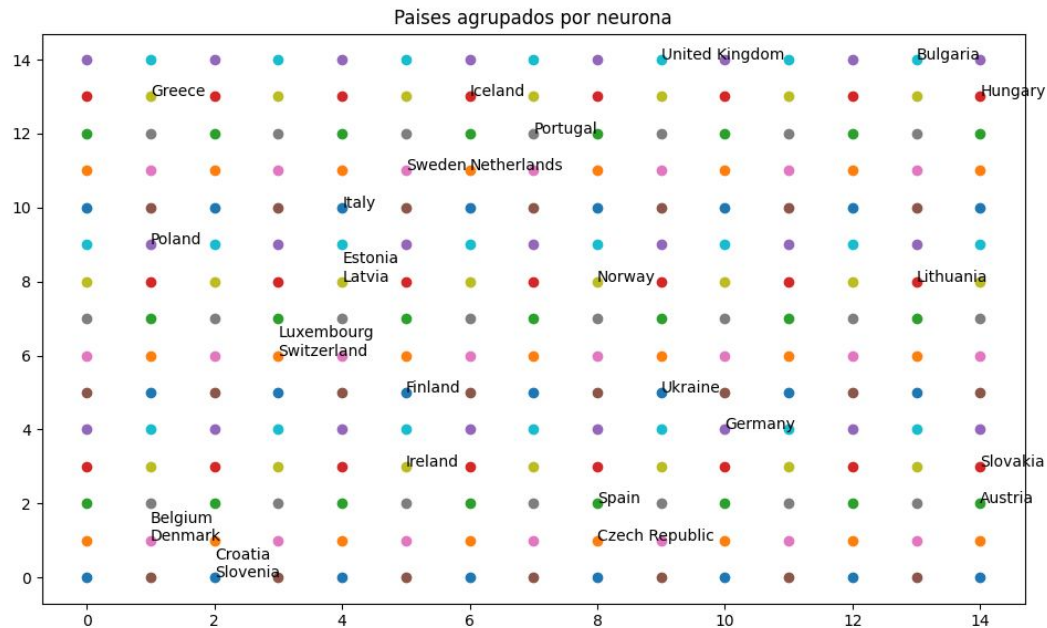
Ejercicio 1a - Resultados

10x10 neuronas, 3000 iteraciones, eta= 0.0001



Ejercicio 1a - Resultados

15x15 neuronas, 3000 iteraciones, eta= 0.0001

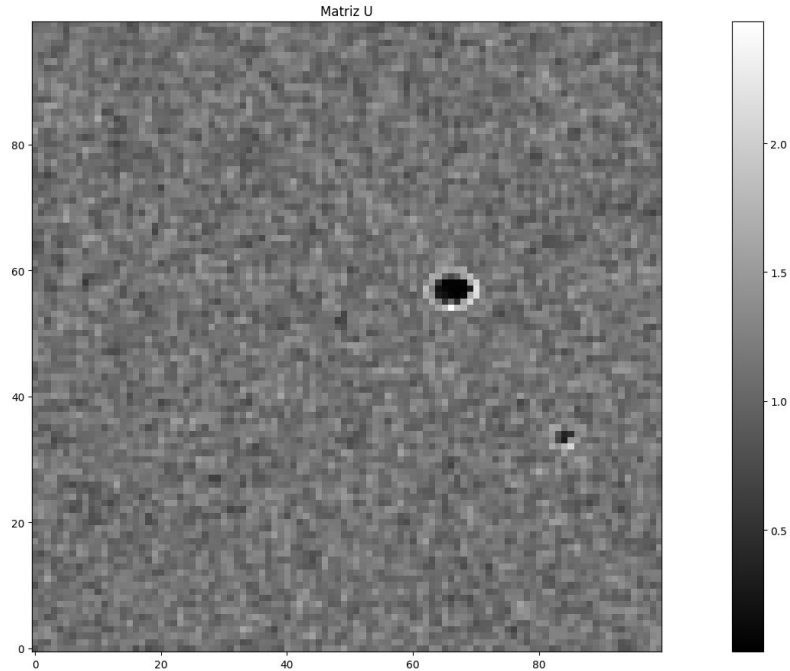




Kohonen - Análisis de dataset de COVID Argentina

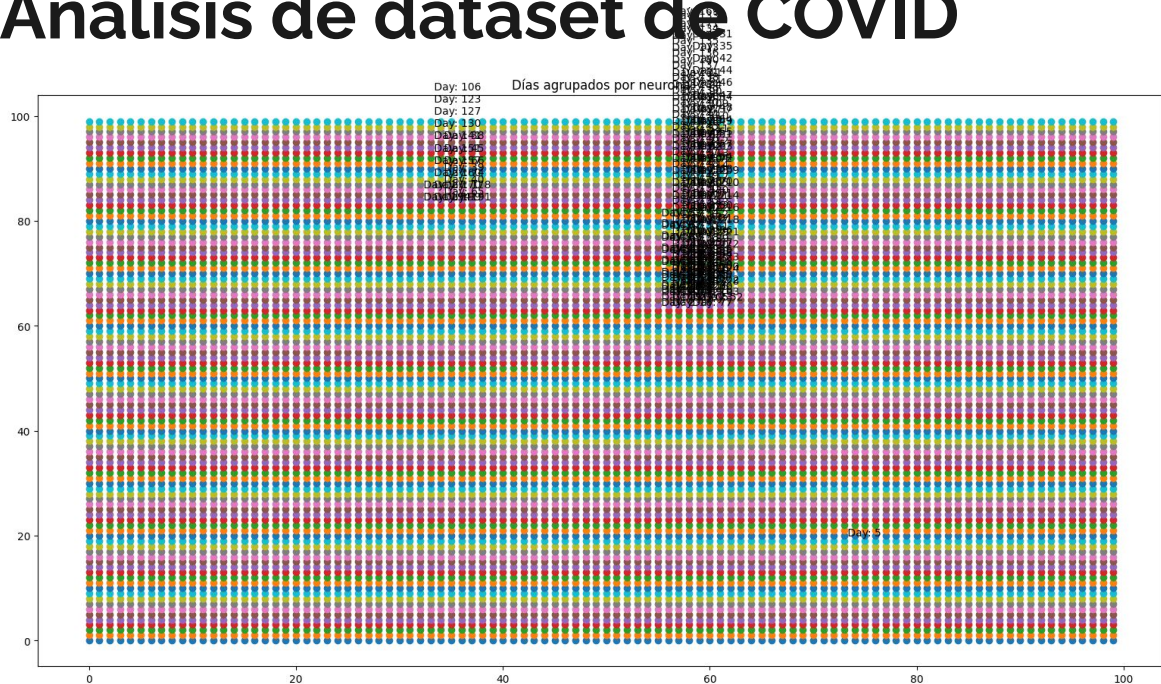
- Datos de Argentina extraídos de <https://ourworldindata.org/coronavirus-source-data>
- Los datos muestran:
 - **Total de casos, nuevos casos, total de muertes, nuevas muertes, total de casos por millón de habitantes, nuevos casos por millón de habitantes, total de muertes por millón de habitantes, nuevas muertes por millón de habitantes**
- Se utilizaron 100 neuronas
- Se utilizó un $\text{ETA} = 0.0001$
- Se buscó la agrupación por similitud entre cada día

Kohonen - Análisis de dataset de COVID Argentina



Kohonen - Análisis de dataset de COVID Argentina

- Notamos que los últimos 150 días son MUY similares y están en neuronas muy cercanas (Ver `covid_analisis.md` en el repo)
- Los fines de semana están siempre en neuronas distintas que los días de semana más próximos



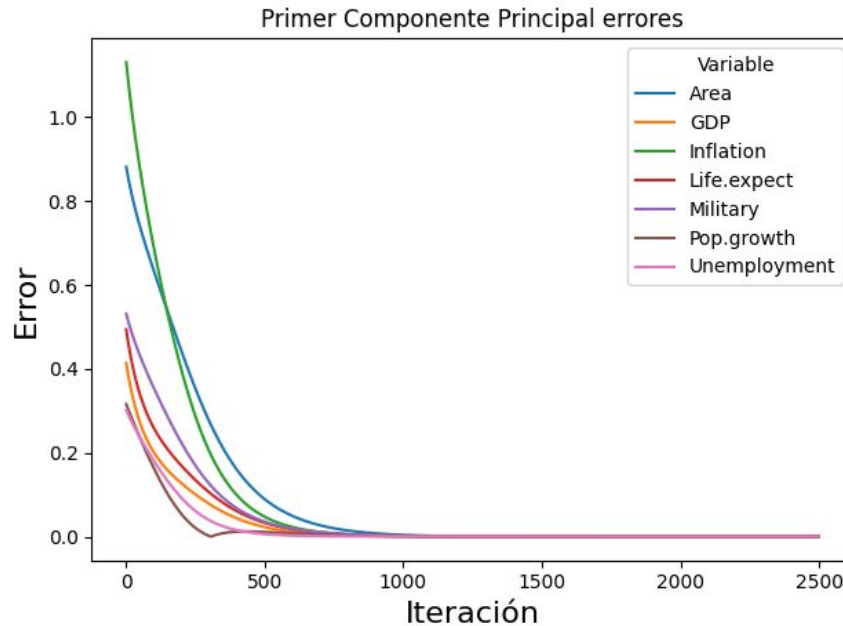


Ejercicio 1

- b) Implementar una red neuronal utilizando la regla de Oja para resolver los siguientes problemas:
- 1) Calcular la primer componente principal para este conjunto de datos.
 - 2) Interpretar el resultado de la primer componente.
 - 3) Comparar el resultado del ejercicio 1b2 con el resultado de calcular la primer componente principal con una librería.

Ejercicio 1b - Resultados

2500 iteraciones y $\eta = 0.0001$





Ejercicio 1b - Resultados

2500 iteraciones y eta= 0.0001

Primer Componente Principal con Oja

[0.12558883, -0.50044312, 0.40722222, -0.48302086, 0.18751431, -0.47555232, 0.27130766]

Primer Componente Principal con PCA

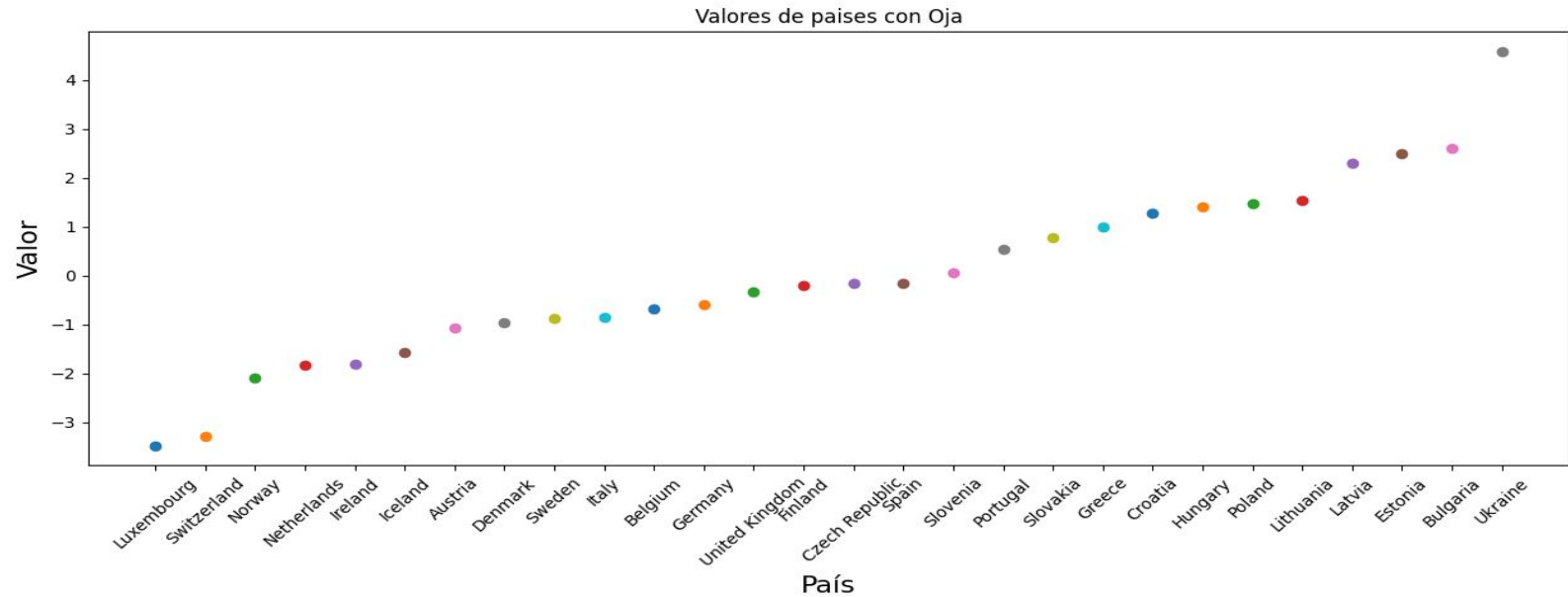
[0.1248739, -0.50050586, 0.40651815 -0.48287333, 0.18811162, -0.47570355, 0.27165582]

Errores finales

[7.14927270e-04, 6.27390556e-05, 7.04061995e-04, 1.47536366e-04, 5.97303936e-04, 1.51231918e-04, 3.48155837e-04]

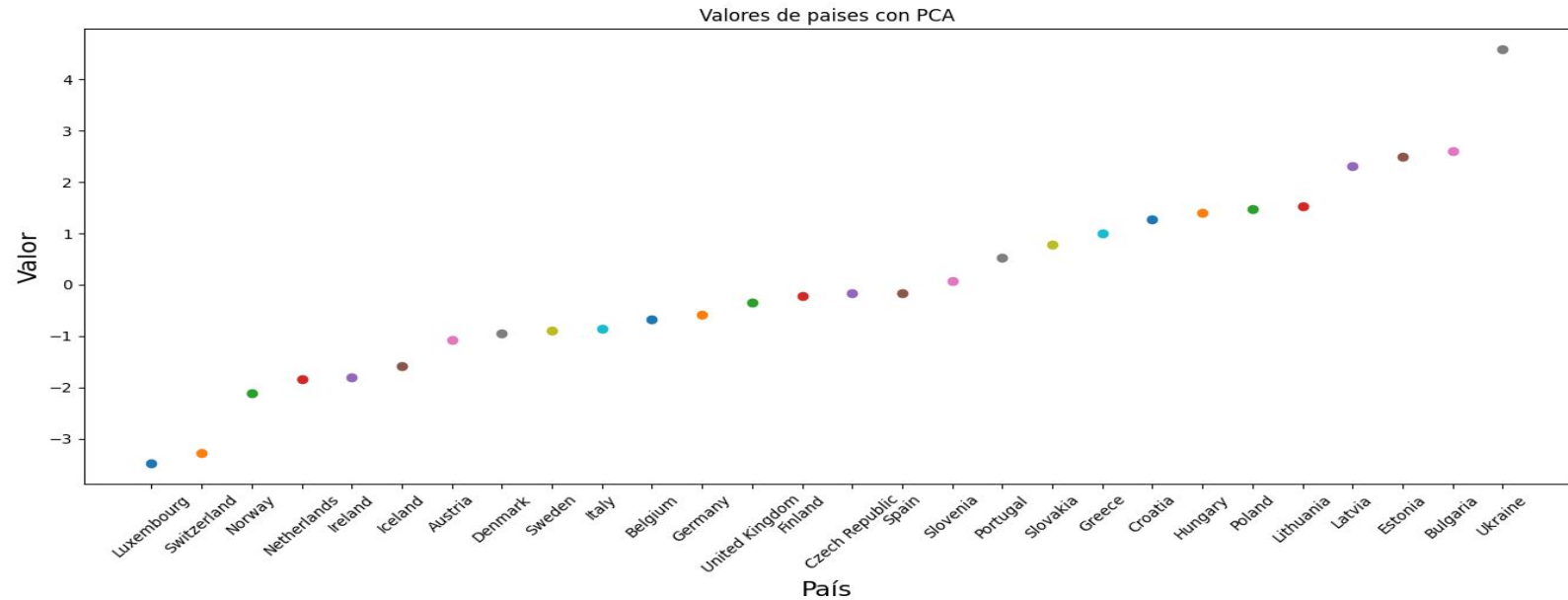
Ejercicio 1b - Resultados

2500 iteraciones y $\eta = 0.0001$



Ejercicio 1b - Resultados

2500 iteraciones y $\eta = 0.0001$





Ejercicio 1b - Resultados

2500 iteraciones y $\eta = 0.0001$

| | | |
|------------|---------------------------|---------------------------|
| Austria | OJA: -1.081052518665592 | PCA: -1.0817476612095884 |
| Belgium | OJA: -0.6812358585210633 | PCA: -0.6810940686338388 |
| Bulgaria | OJA: 2.6091847681582947 | PCA: 2.609878816420479 |
| Croatia | OJA: 1.2679219941856519 | PCA: 1.2701488485858363 |
| Czech Rep. | OJA: -0.16794233439188117 | PCA: -0.1672094937905082 |
| Denmark | OJA: -0.955460197020773 | PCA: -0.9551907971097163 |
| Estonia | OJA: 2.4874856096933815 | PCA: 2.4877352181041092 |
| Finland | OJA: -0.21002490685879227 | PCA: -0.21056315636343792 |
| Germany | OJA: -0.5917769103661773 | PCA: -0.5923936529244794 |
| Greece | OJA: 0.9975535017333447 | PCA: 1.0004719569396432 |
| Hungary | OJA: 1.3966787469644157 | PCA: 1.396898312783373 |
| Iceland | OJA: -1.5821898010237587 | PCA: -1.5837197024631102 |
| Ireland | OJA: -1.8092846011604042 | PCA: -1.8089176109391967 |
| Italy | OJA: -0.8529909769704404 | PCA: -0.8532239623948354 |

| | | |
|-------------|---------------------------|---------------------------|
| Latvia | OJA: 2.306277836280905 | PCA: 2.306059408593168 |
| Lithuania | OJA: 1.5301071666612058 | PCA: 1.53009990995271 |
| Luxembourg | OJA: -3.4778183275001178 | PCA: -3.478434956704635 |
| Netherlands | OJA: -1.8406987861259885 | PCA: -1.8400534050753583 |
| Norway | OJA: -2.1065306256946297 | PCA: -2.106510833605665 |
| Poland | OJA: 1.4724220185257635 | PCA: 1.4717738276291163 |
| Portugal | OJA: 0.5255668736847512 | PCA: 0.5264933349515398 |
| Slovakia | OJA: 0.7829417440942292 | PCA: 0.7829659710814738 |
| Slovenia | OJA: 0.06581819415995376 | PCA: 0.06754337538258681 |
| Spain | OJA: -0.16297627177377694 | PCA: -0.16376695684389564 |
| Sweden | OJA: -0.8838573242200324 | PCA: -0.8851053096517574 |
| Switzerland | OJA: -3.282589850313138 | PCA: -3.281586128815986 |
| Ukraine | OJA: 4.584992869083376 | PCA: 4.5802680667693805 |
| UK | OJA: -0.34052203261863423 | PCA: -0.34081935066733593 |

Ejercicio 2

Construir patrones de letras del abecedario utilizando 1 y -1 y matrices de 5×5. Por ejemplo, con la matriz:

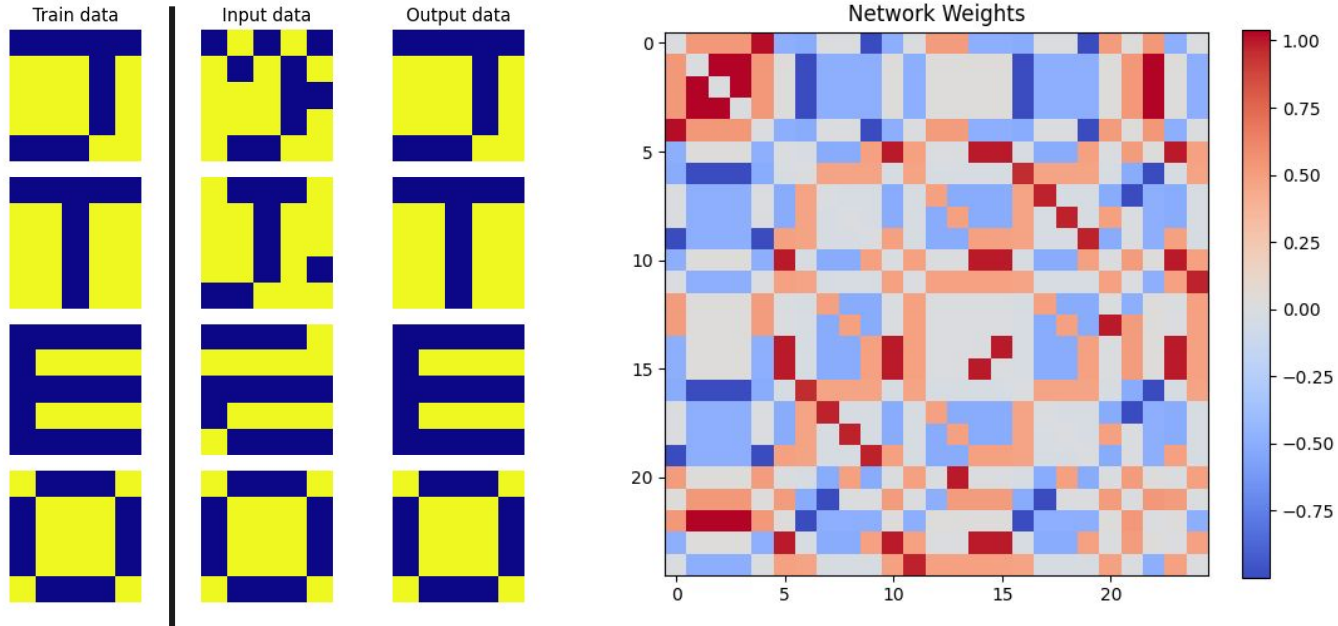
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 \end{pmatrix}$$

, puede dibujarse la letra J:

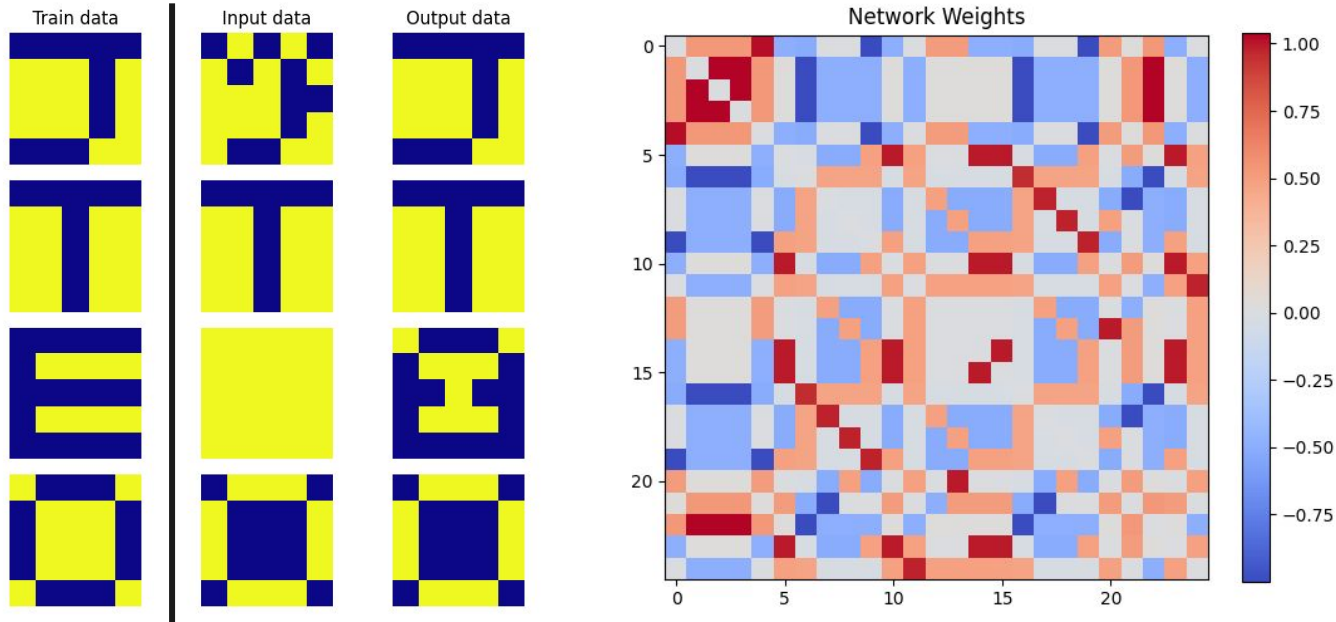
```
* * * * *
      *
      *
      *
*      *
* * *
```

- Almacenar 4 patrones de letras. Realizar un programa que aplique el modelo de Hopfield para asociar matrices ruidosas de 5×5 con los patrones de las letras almacenadas. Los patrones de consulta deben ser alteraciones aleatorias de los patrones originales. Mostrar los resultados que se obtienen en cada paso hasta llegar al resultado final.
- Ingresa un patrón muy ruidoso e identificar un estado espúreo.

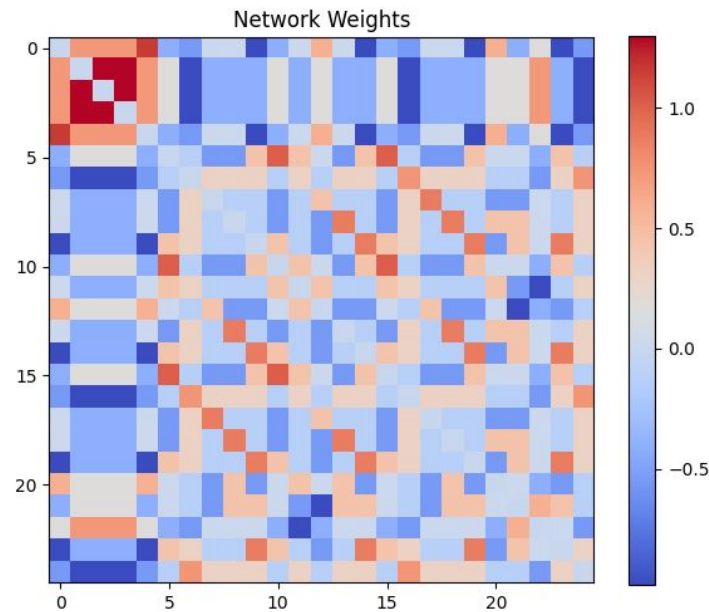
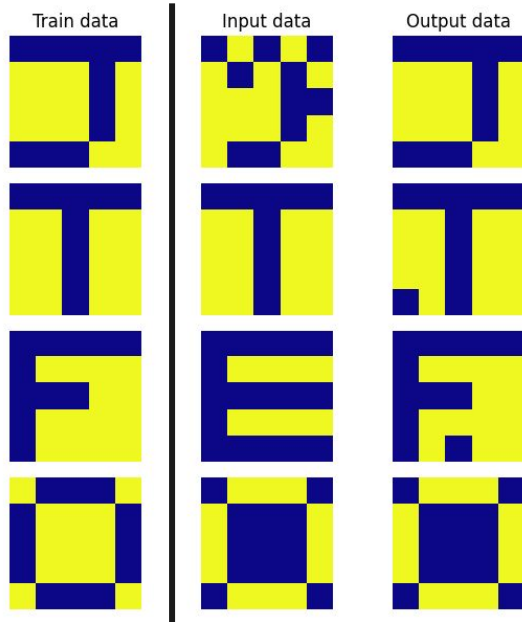
Ejercicio 2 - Resultados



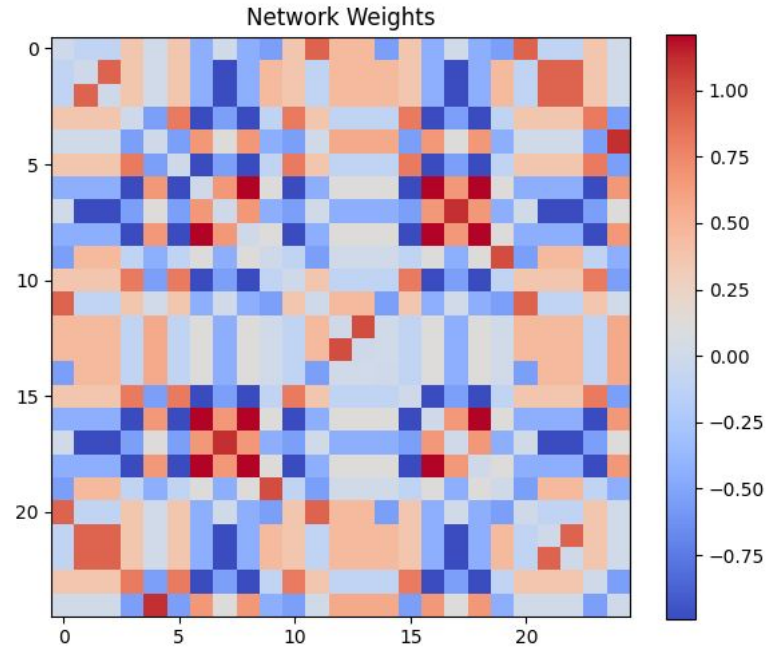
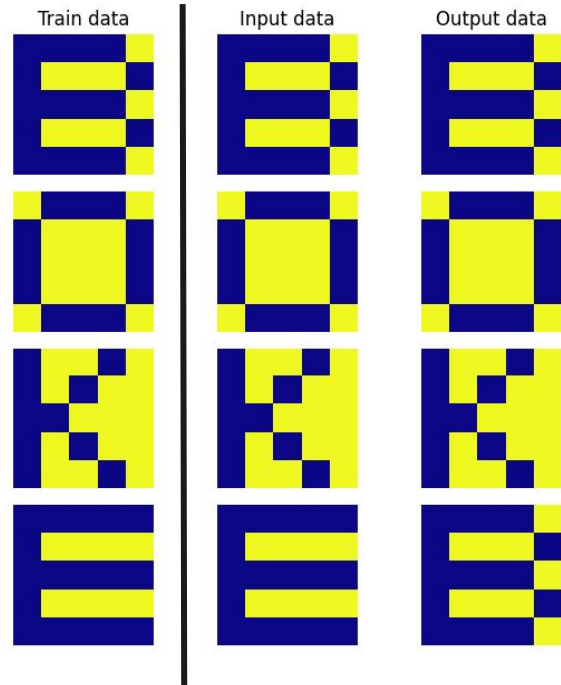
Ejercicio 2 - Resultados estados espurios



Ejercicio 2 - Resultados con patrones similares de entrenamiento



Ejercicio 2 - Resultados





Conclusiones

- Kohonen:
 - Con una cantidad grande de información, se pueden notar los clusters agrupando los datos de una manera más marcada
 - Es importante la cantidad de neuronas utilizadas, dependiendo de lo que se este analizando y el eta
- Oja:
 - Se puede observar claramente la convergencia de esta regla con los errores obtenidos.
 - Es muy importante el eta utilizado, ya que puede no converger.
- Hopfield:
 - Eligiendo 4 letras con patrones similares, por ejemplo J, T, F, E, la exactitud de la resolución ante un mínimo ruido disminuye y se consiguen varios estados espurios