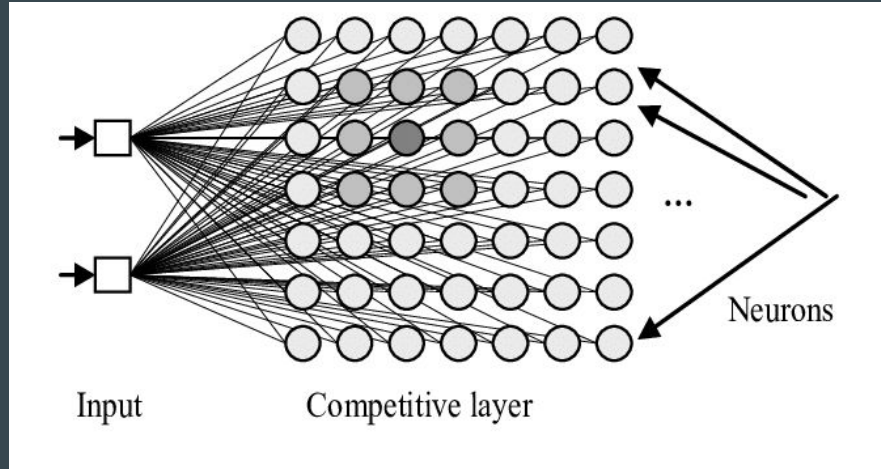


Sistemas de Inteligencia Artificial:

TP4 - Aprendizaje no Supervisado



Ejercicio 1.a: Red de Kohonen

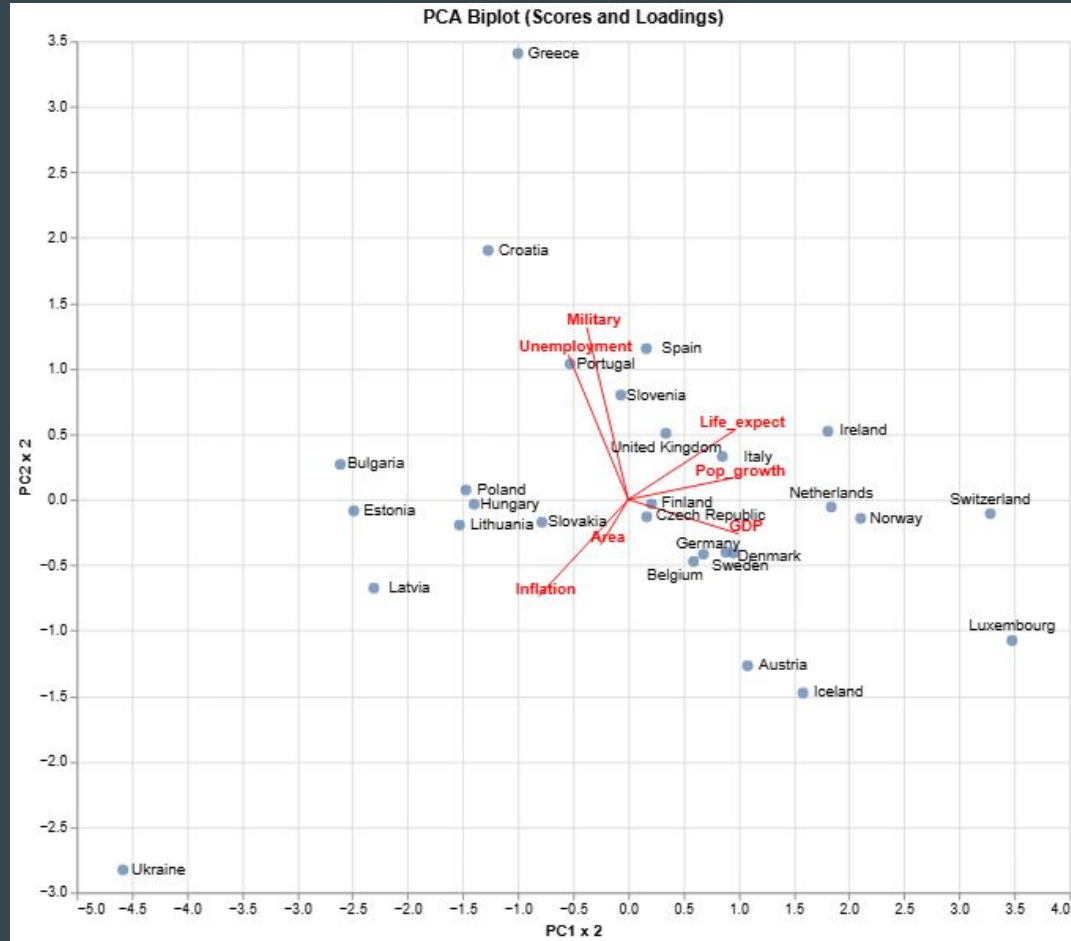
Refrescando el dataset...

	Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0	Austria	83871	41600	3.5	79.91	0.80	0.03	4.2
1	Belgium	30528	37800	3.5	79.65	1.30	0.06	7.2
2	Bulgaria	110879	13800	4.2	73.84	2.60	-0.80	9.6
3	Croatia	56594	18000	2.3	75.99	2.39	-0.09	17.7
4	Czech Republic	78867	27100	1.9	77.38	1.15	-0.13	8.5

Ejercicio 1.a: Red de Kohonen

Refrescando las categorías
por países...

PC1 y PC2 representa un
63.06% de variabilidad de
los datos

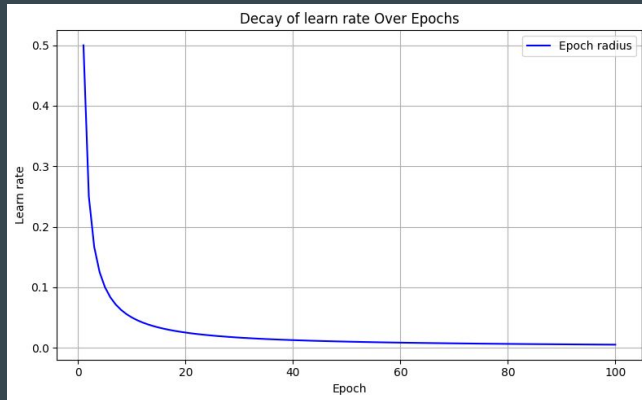


Ejercicio 1.a: Red de Kohonen

Para estos problemas se permite utilizar funciones de decrecimiento para el learn_rate y para el radio, las funciones que se usan son las siguientes:

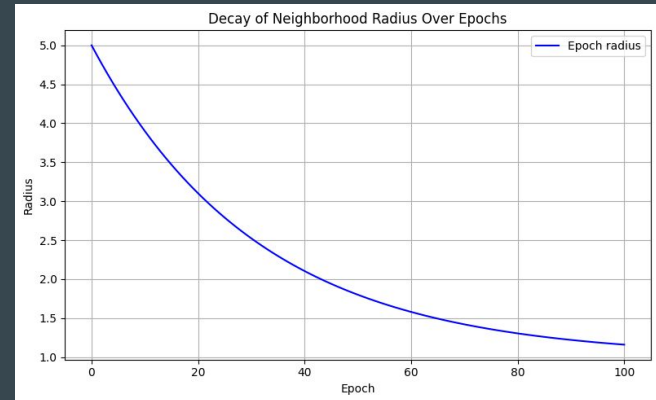
$$\eta(epoch) = \frac{\eta_0}{epoch}$$

$$r(epoch) = 1 + (r_0 - 1) \cdot \exp\left(-\frac{epoch \cdot \ln(r_0)}{\max_epochs}\right)$$



En ejemplos:

max_epochs = 100
n0 = 0.5
r0 = 5



Ejercicio 1.a: Red de Kohonen - Clasificación por neurona

Params:

```
np.random.seed(31)
```

```
K = 4
```

```
max_epochs = 100
```

```
use_weights_from_inputs = true
```

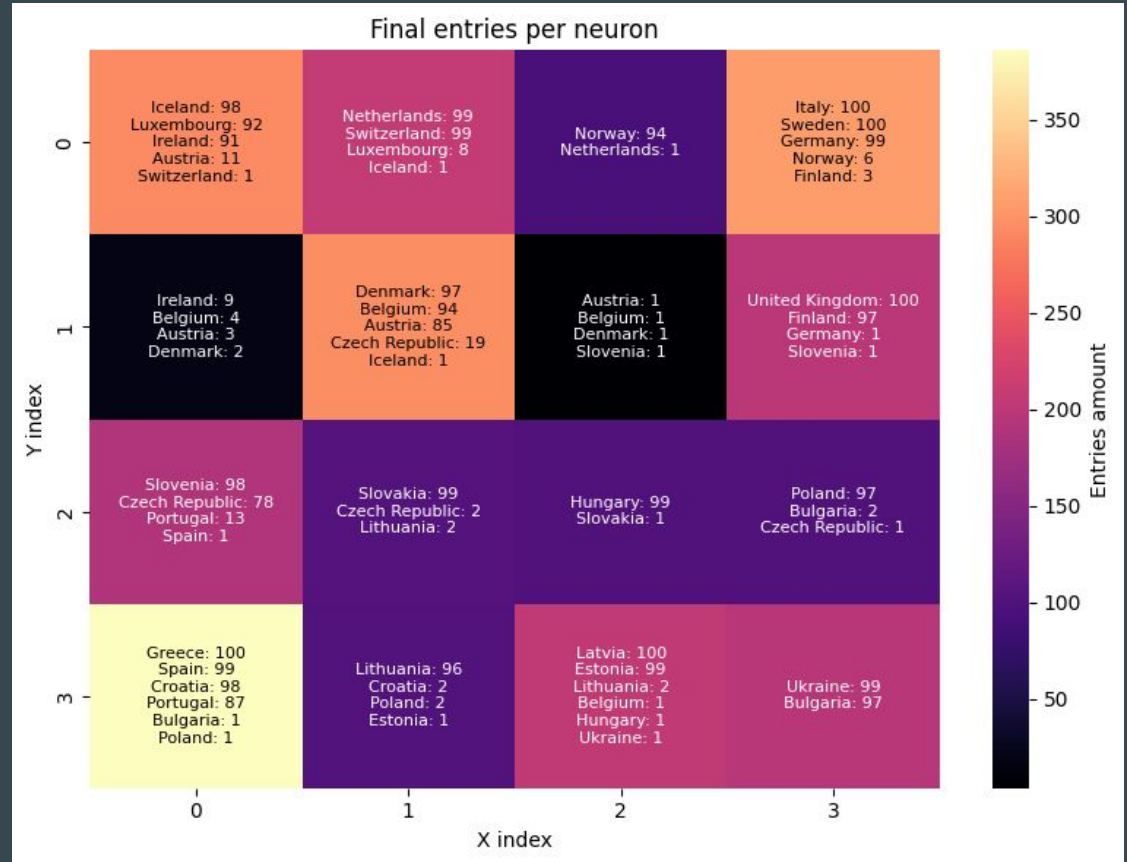
```
ini_radius = k / 2
```

```
decrease_radius = true
```

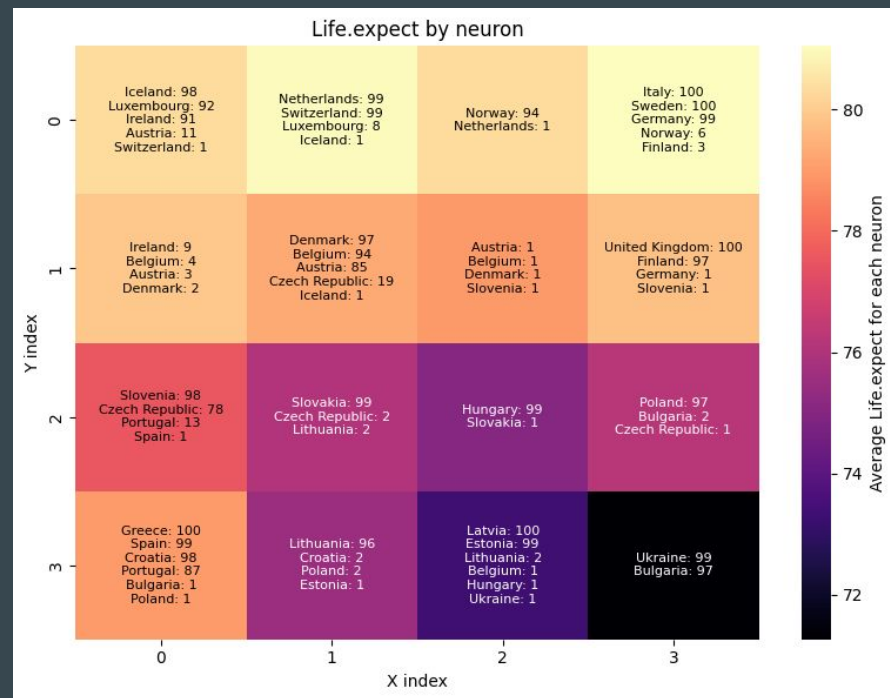
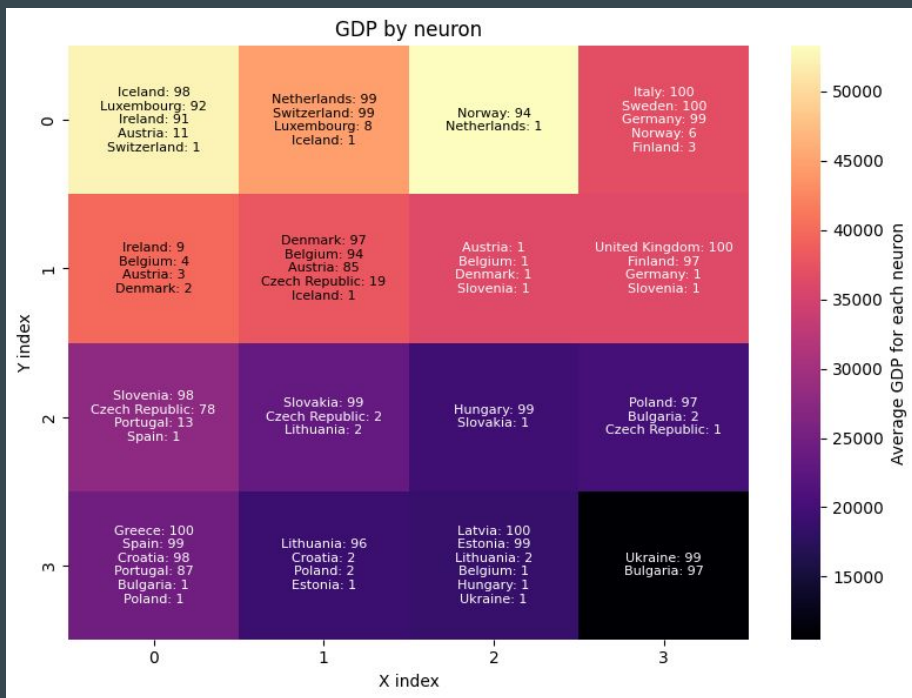
```
learn_rate = 0.5
```

```
decrease_learn_rate = true
```

(Se dejarán estos parámetros fijos para las siguientes comparaciones, cuando se quiera probar algún hiperparámetro sólo se variará ese)

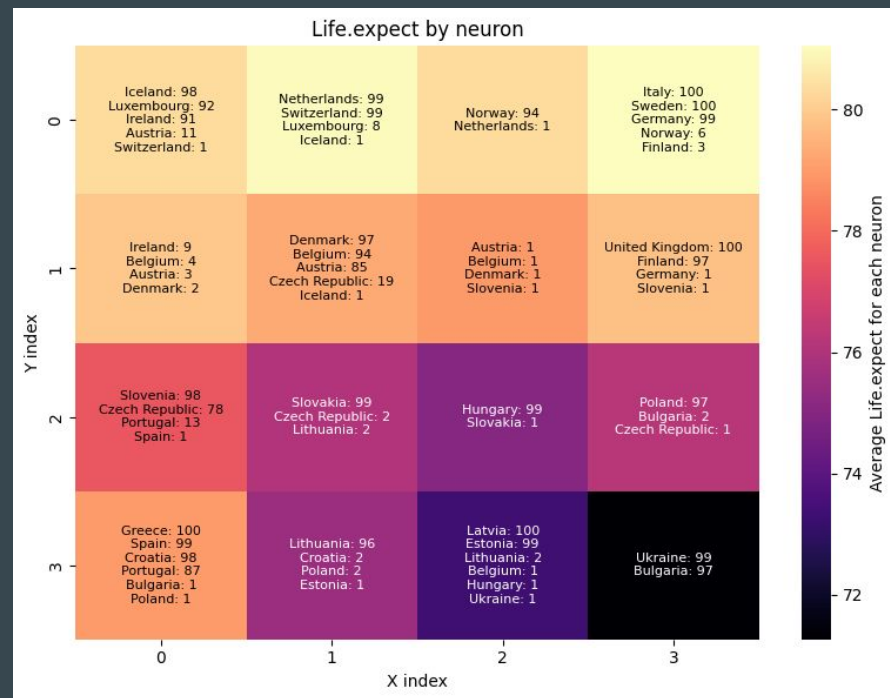
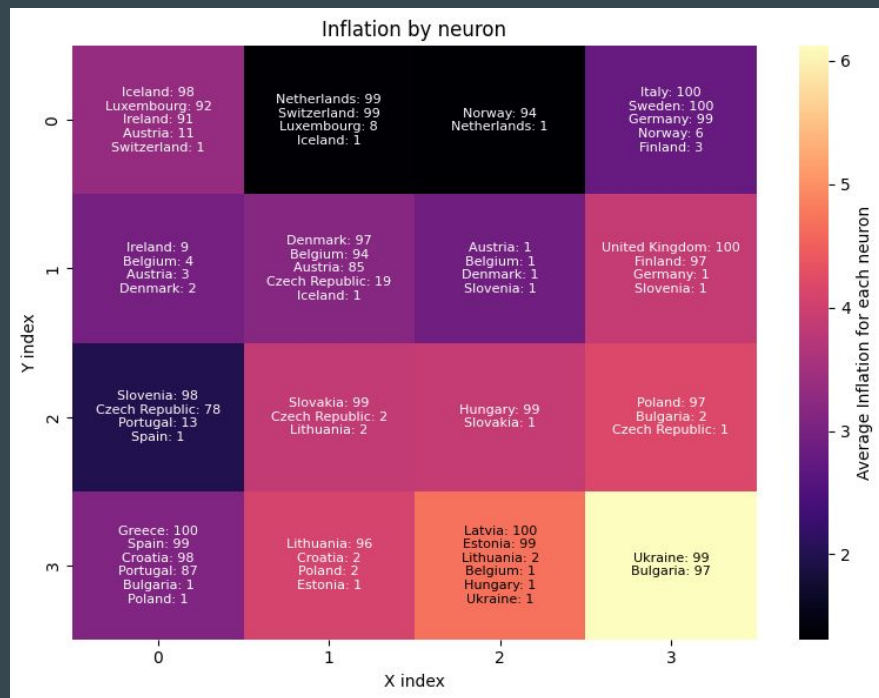


Ejercicio 1.a: Red de Kohonen - Clasificación por neurona



Se observa que dónde más se activan las neuronas por GDP, también es dónde más se activan por Life expect

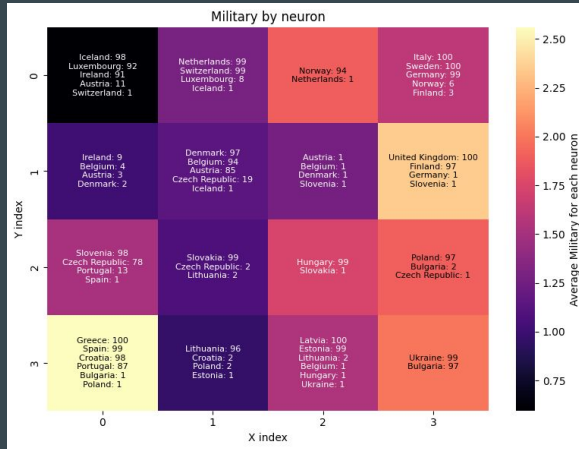
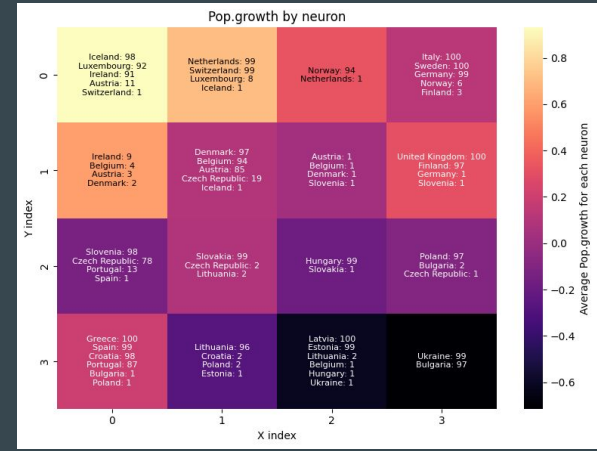
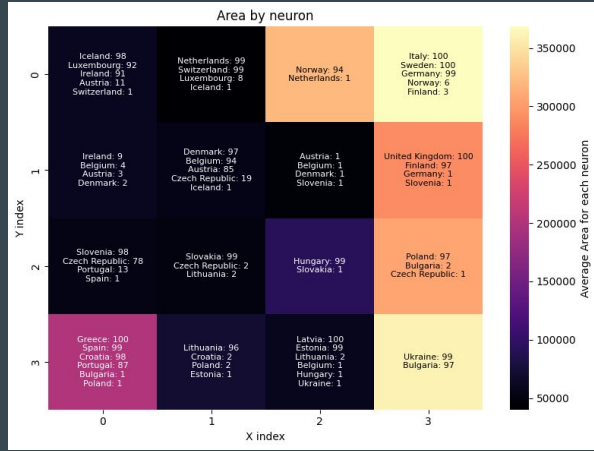
Ejercicio 1.a: Red de Kohonen - Clasificación por neurona



Se observa que dónde más se activan las neuronas Inflation, menos se activan por Life expect

Ejercicio 1.a: Red de Kohonen - Clasificación por neurona

Un vistazo
a las otras
variables



Ejercicio 1.a: Red de Kohonen - Épocas

Se define una medida de similitud para ver cuánto cambiaron los pesos de las N neuronas desde una época a la siguiente:

$$\%Sim_{i+1} = \frac{1}{N} \sum_{j=1}^N \frac{w_j^{(i)} \cdot w_j^{(i+1)}}{\|w_j^{(i)}\| \cdot \|w_j^{(i+1)}\|} \cdot 100$$

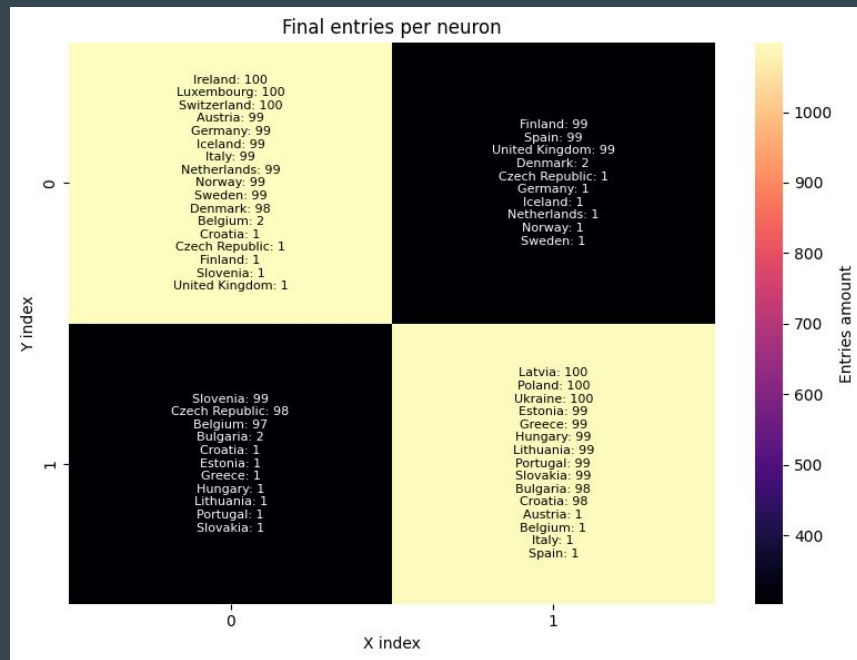
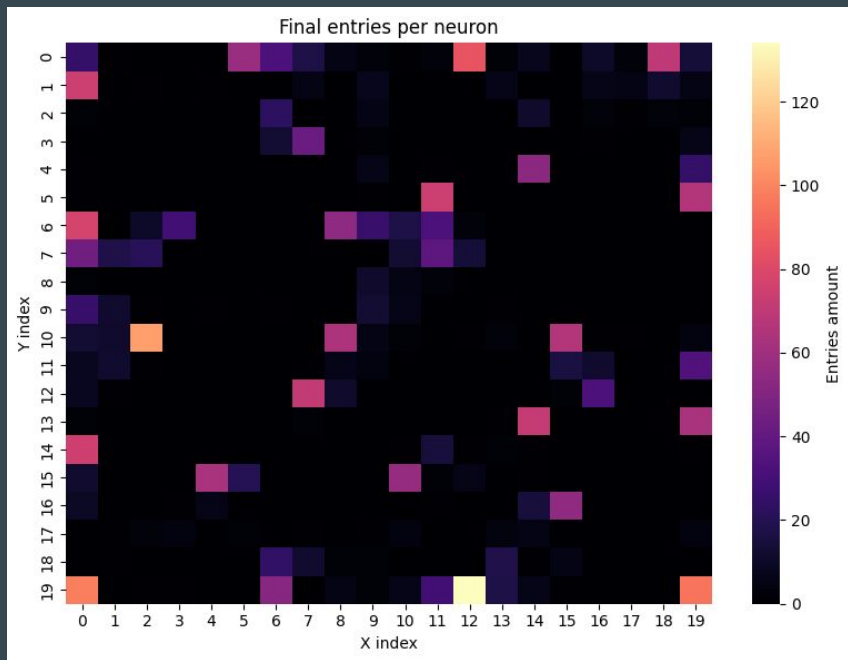
Probando para 100 seeds distintas pero manteniendo constantes el resto de parámetros se obtuvo que los pesos convergen a un 99.9% de similitud de su anterior época en la época promedio de:

$$e = 6.98 \pm 1.65$$

Es decir, los pesos en este problema convergen de manera bastante rápida

Ejercicio 1.a: Red de Kohonen - Distintos K

¿Qué pasa cuando tenemos un K muy grande? ¿Y un K muy chico?

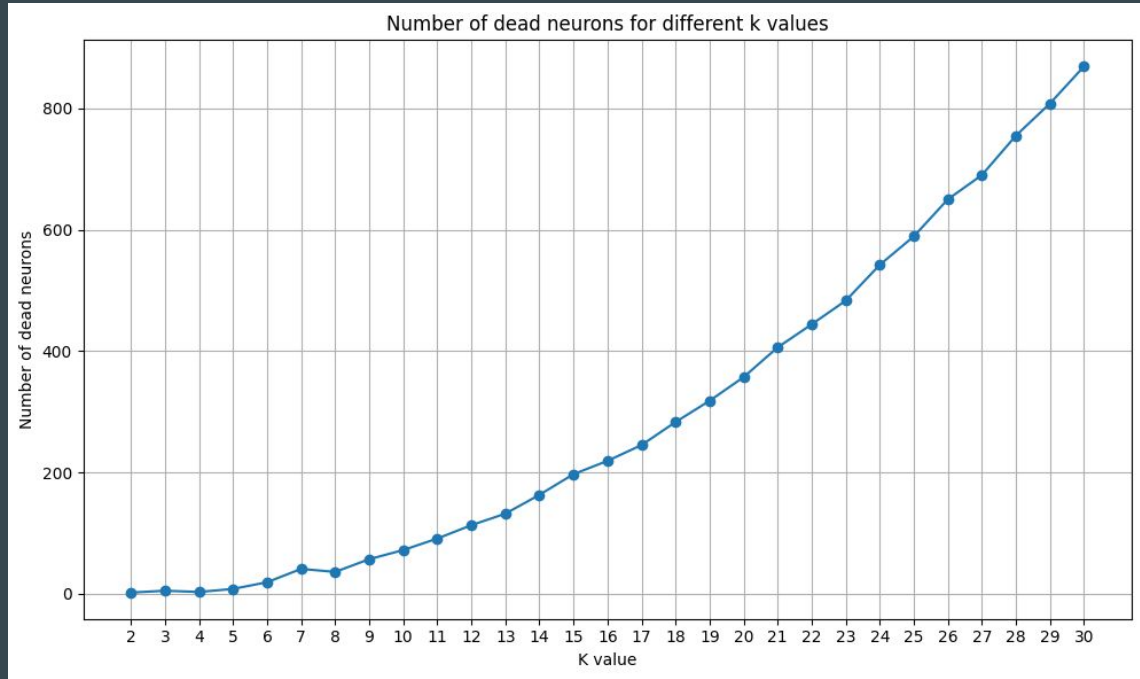


Quedan datos muy separados o muy agrupados (no se aprecian agrupamientos significativos de datos)

Ejercicio 1.a: Red de Kohonen - Distintos K

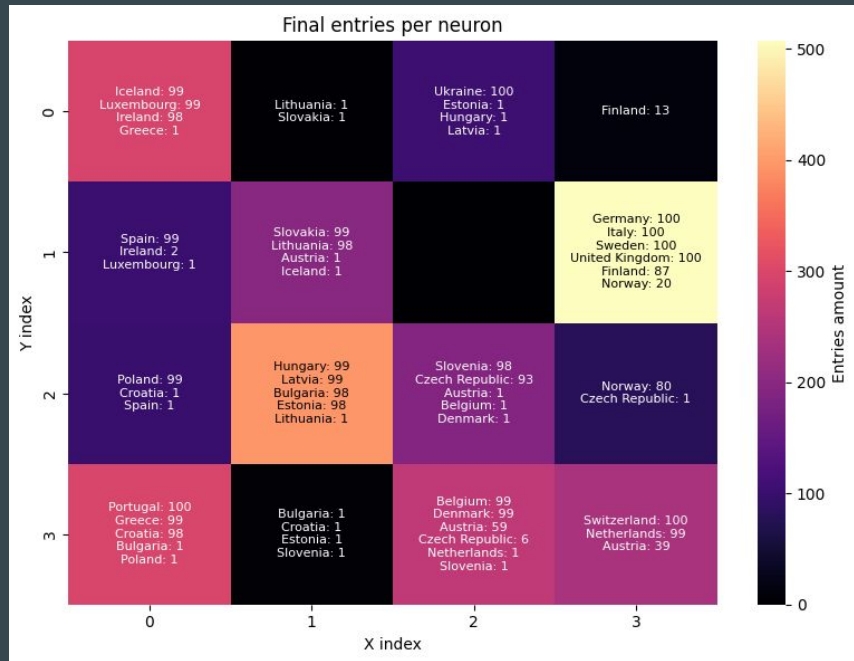
Se puede observar que a mayor K en este dataset, hay un crecimiento exponencial de la cantidad de neuronas muertas.

Es importante tomar un K para un dado dataset que genere una cantidad pequeña de neuronas muertas. Para apreciar mejor las relaciones entre mis datos.

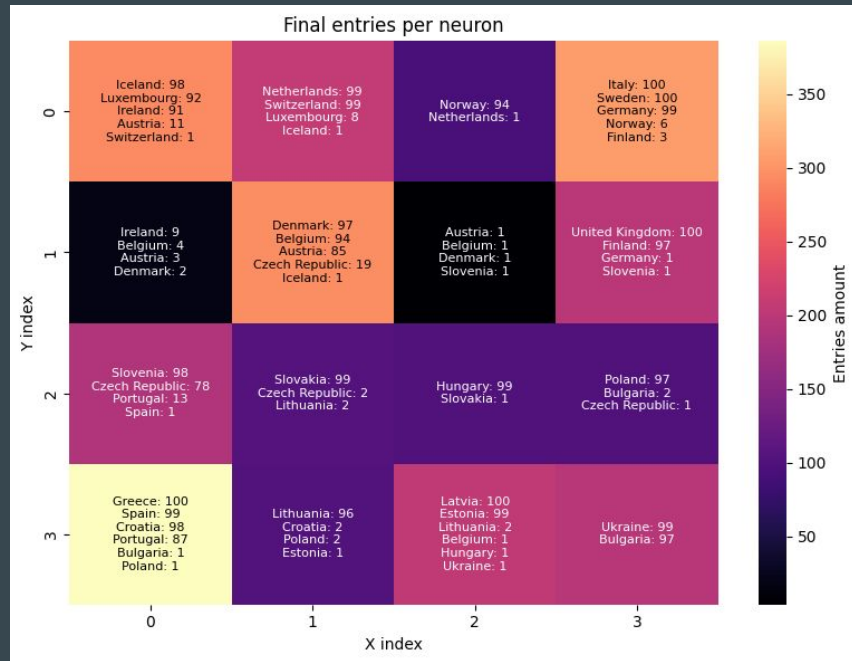


Ejercicio 1.a: Red de Kohonen - Elección de radio inicial

$$R_0 = 1$$

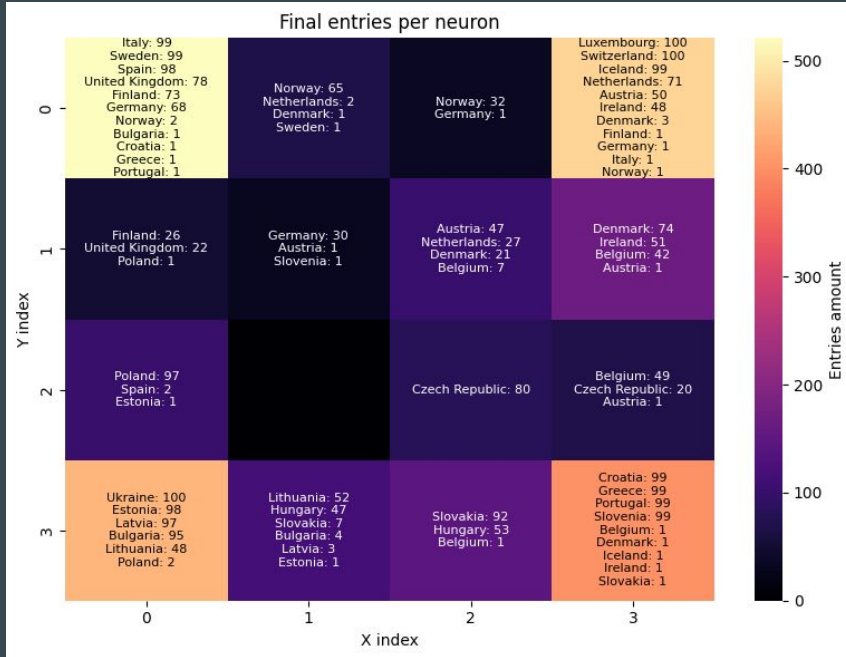


$$R_0 = 2 = (k/2)$$

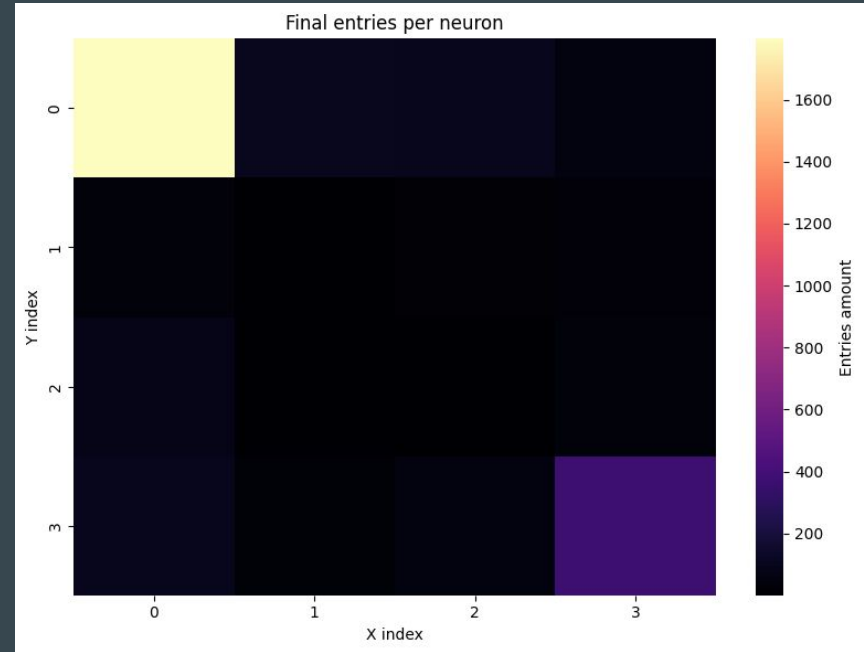


Ejercicio 1.a: Red de Kohonen - Elección de radio inicial

$$R_0 = 3$$



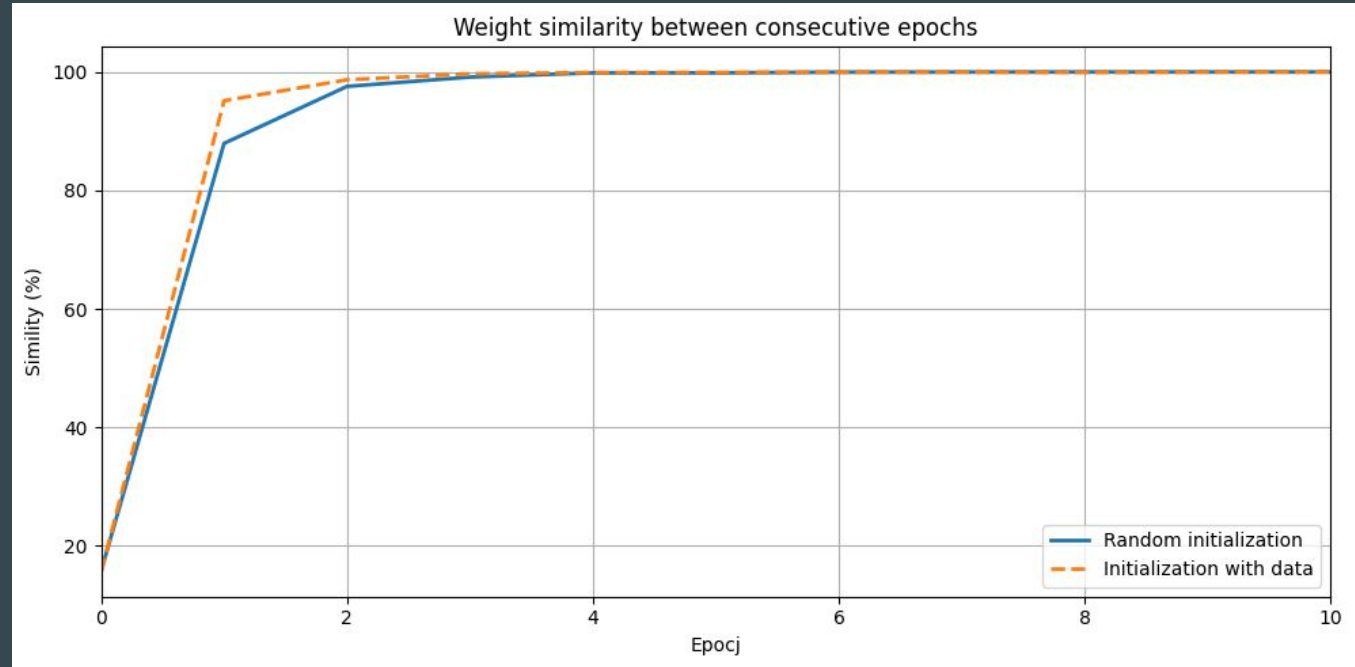
$$R_0 = 16$$



A pesar de que el radio vaya decayendo, la convergencia temprana de los pesos en la red de Kohonen hace que un mal radio inicial dañe irreparablemente la clusterización

Ejercicio 1.a: Red de Kohonen - Inicialización

Usando inicialización de pesos con los datos, se observa una convergencia más temprana aunque no es tan apreciable, tal vez con mayores volúmenes de datos sea más notoria



Ejercicio 1.b: Regla de Oja

Recordando el PC1 calculado en la actividad extra...

$$PC1 = -0.124874 X1 + 0.500506 X2 - 0.406518 X3 + 0.482873 X4 - 0.188112 X5 + 0.475704 X6 - 0.271656 X7$$

Para calcular la similitud de las cargas del PC1 y los pesos hallados por la regla de Oja:

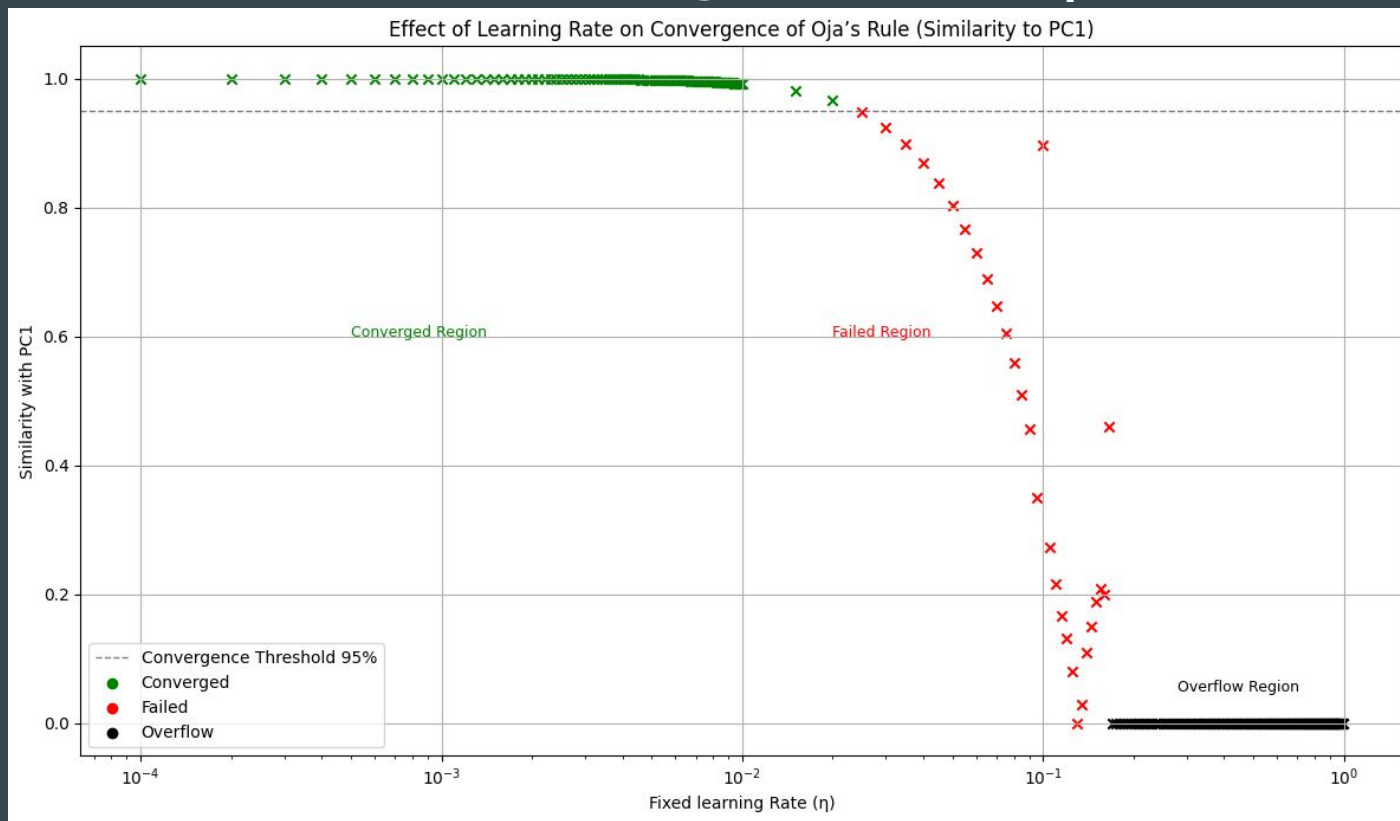
$$v = loadings(PC1)$$

$$w = weights$$

$$similarity = \left| \frac{v \cdot w}{\|v\| \|w\|} \right|$$

Se buscó probar cómo afecta el learn rate (η) a la convergencia de los pesos en las cargas del PC1. Se probó para learn rate fijo, y para learn rate decreciente, tomando un máximo de 1000 épocas.

Ejercicio 1.b: Regla de Oja - η fijo



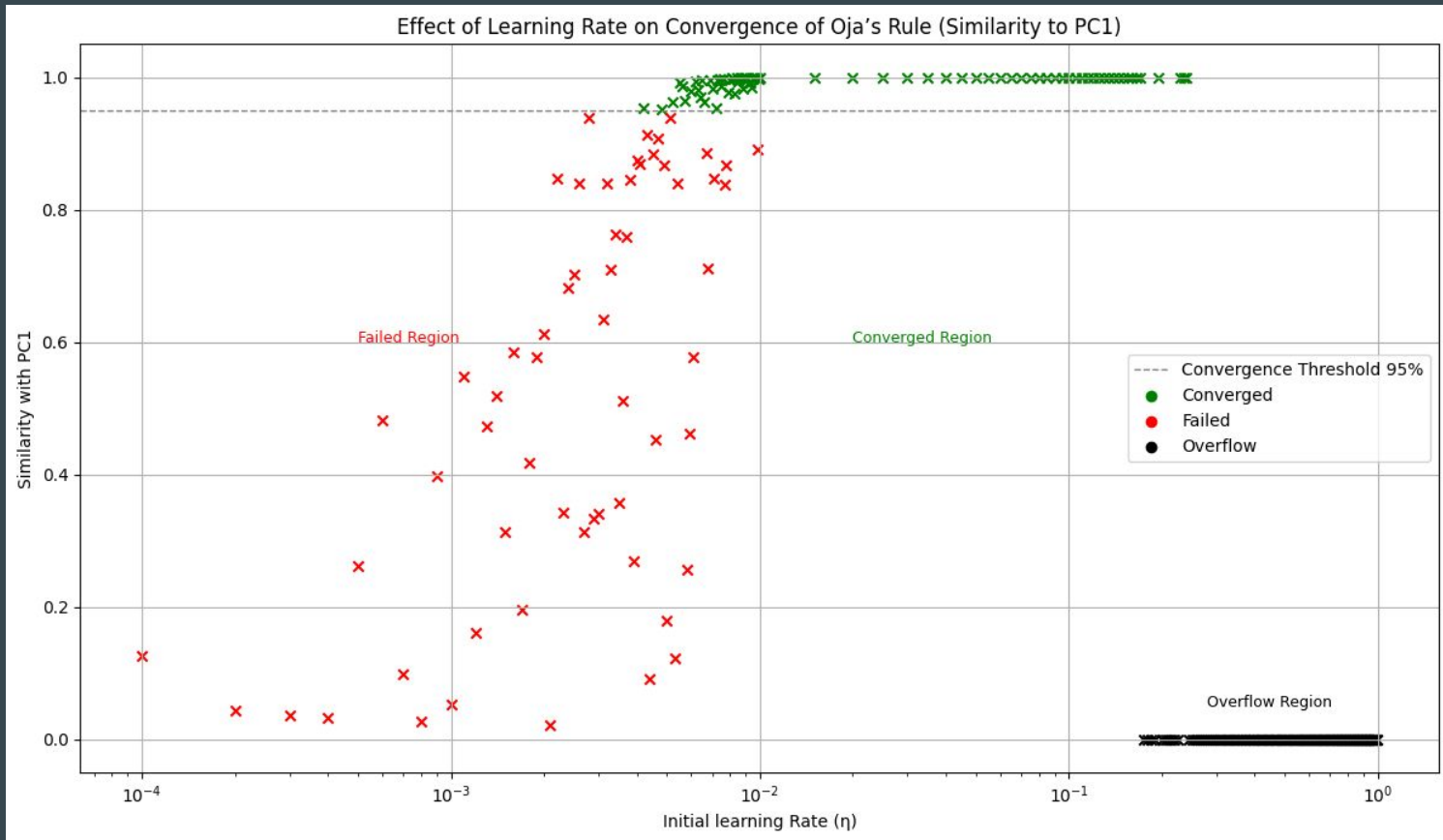
Nota: la Overflow Region es porque empiezan a diverger los pesos a un valor tan alto hasta que da **inf**

Ejercicio 1.b: Regla de Oja - η decreciente

Se define un η_0 y se lo decrece con la fórmula:

$$\eta(epoch) = \frac{\eta_0}{epoch}$$

Ejercicio 1.b: Regla de Oja - η decreciente



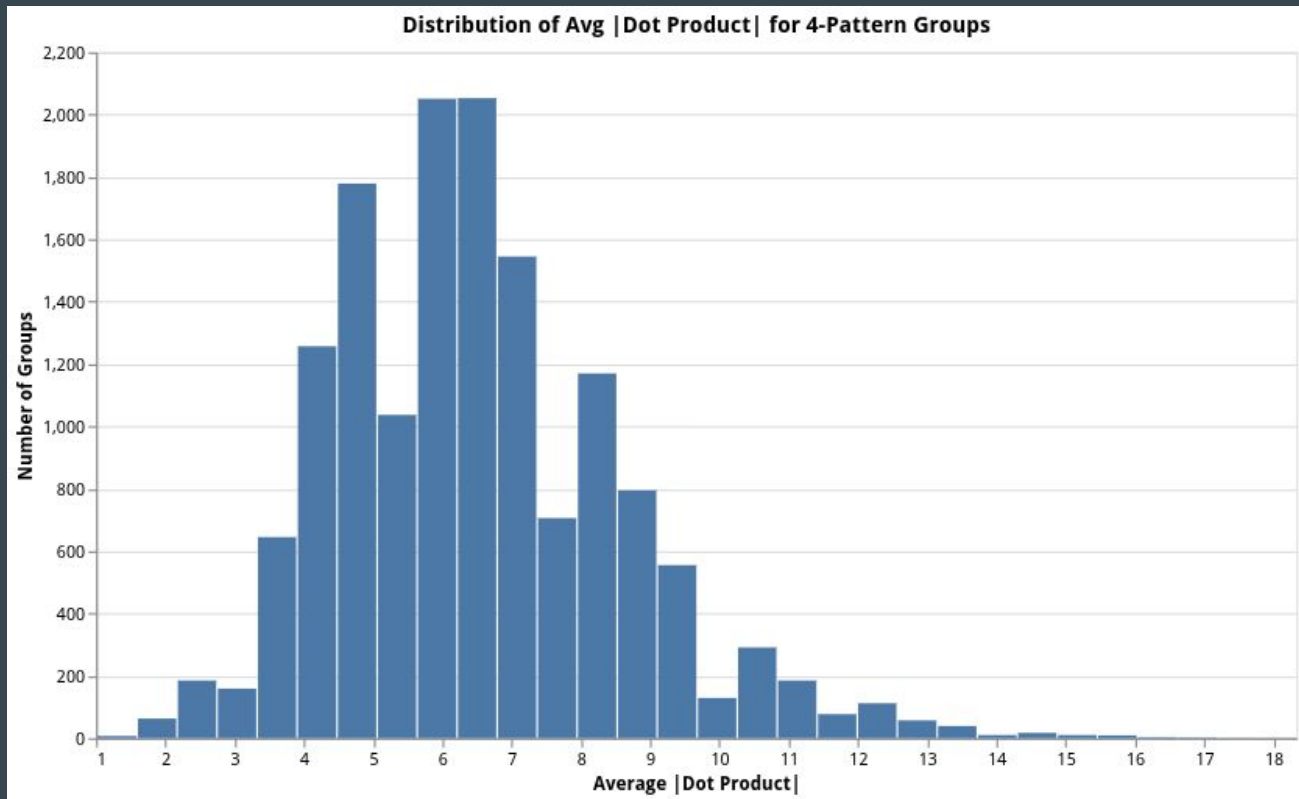
Ejercicio 2: Red de Hopfield

Para este ejercicio se busca crear una red de Hopfield que almacene e identifique 4 patrones de letras. Cada patrón es una representación ASCII de 5x5 de una letra (almacenada en forma de un vector columna de 25 elementos con valor 1 o -1).

Para que esta red funcione adecuadamente se busca que los patrones sean lo más ortogonales posible. Para encontrar qué patrones cumplen esta propiedad se procede a calcular la ortogonalidad de todos los conjuntos posibles de 4 patrones.

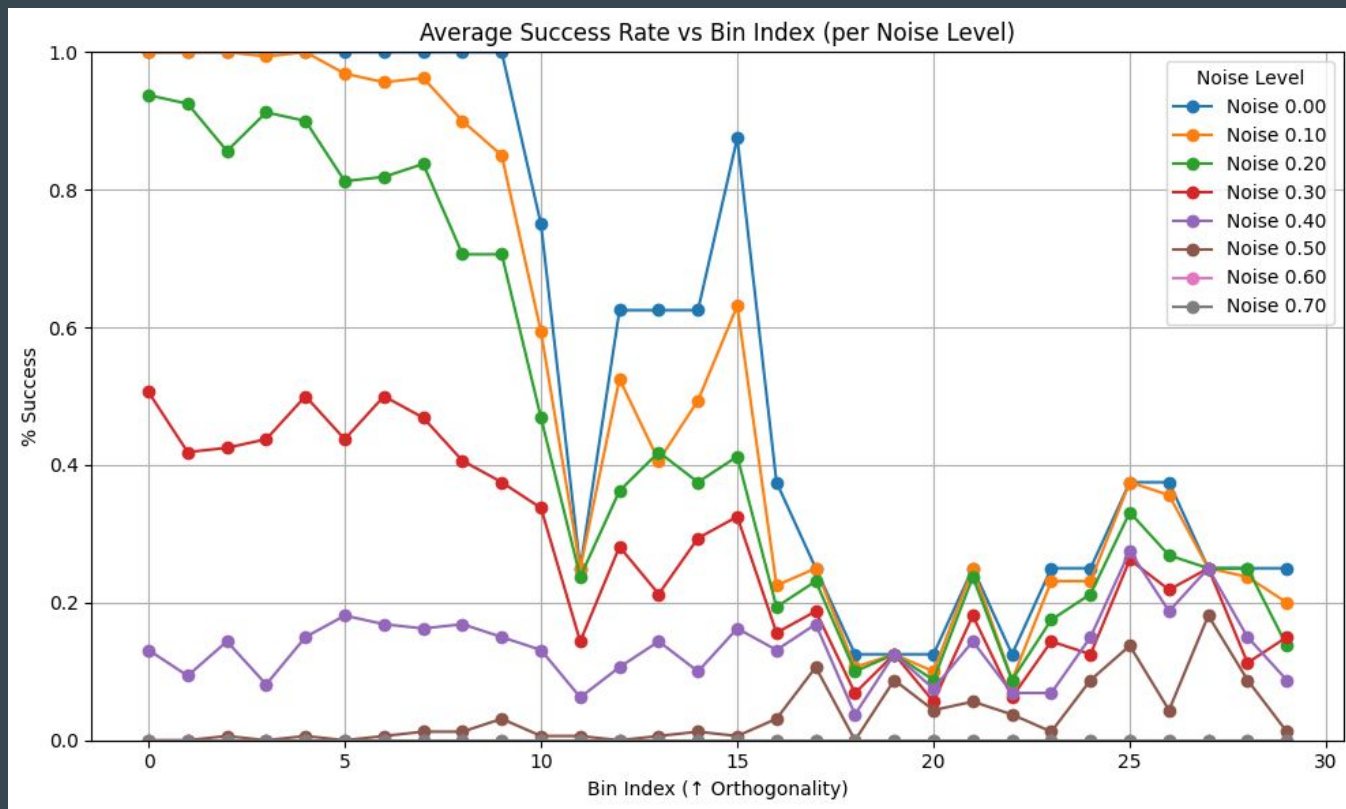
Ortogonalidad

Para cada grupo de patrones se calculó el producto escalar entre cada par de patrones. Luego se agruparon en 30 bins de acuerdo al promedio de esos productos escalares. Con estos bins se graficó un histograma mostrando la cantidad de grupos por bin. Lo que buscamos son patrones de los bins más bajos, es decir los que tienen mejor ortogonalidad.

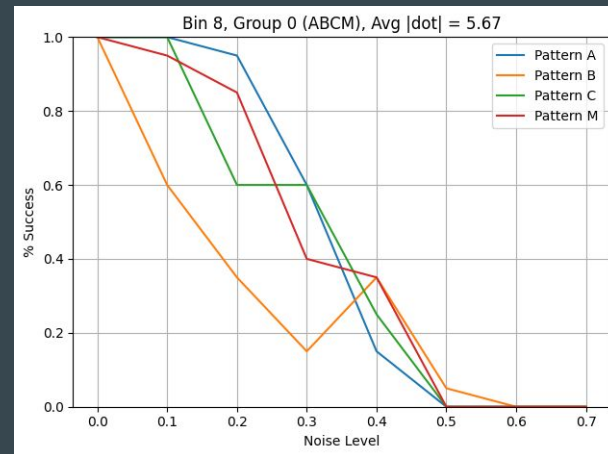
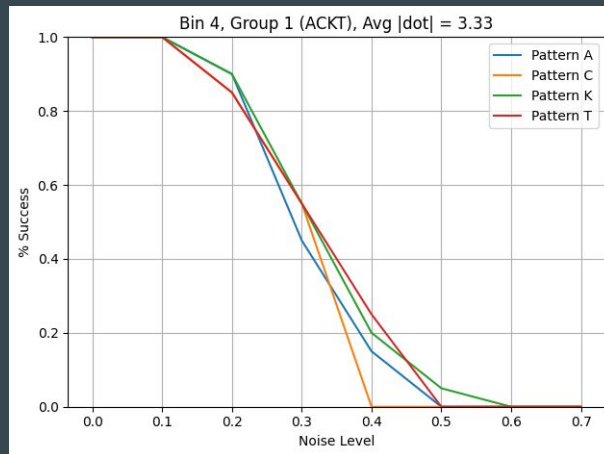
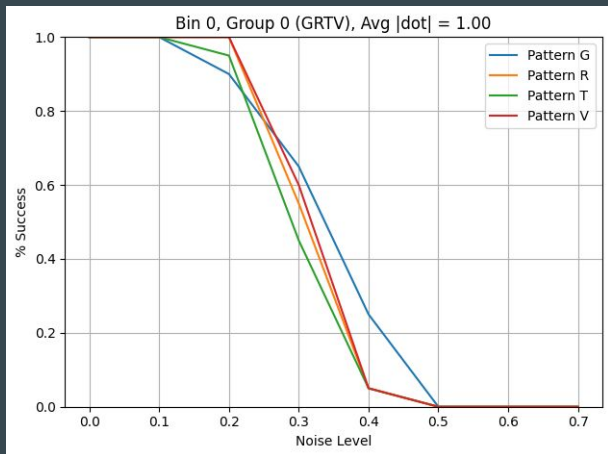


Desempeño ante ruido según ortogonalidad

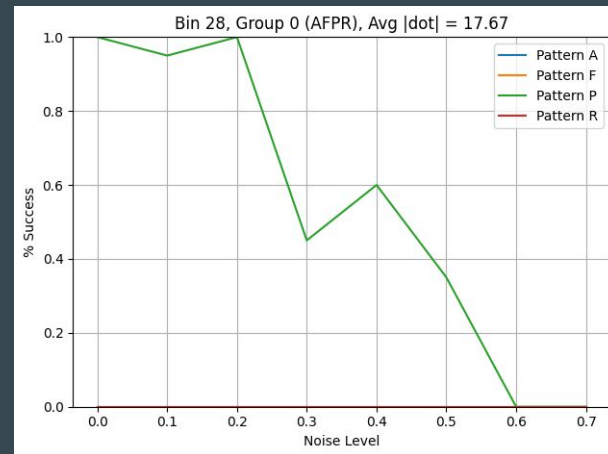
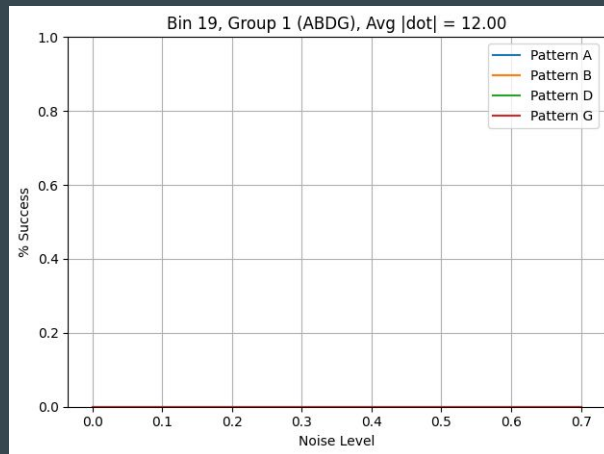
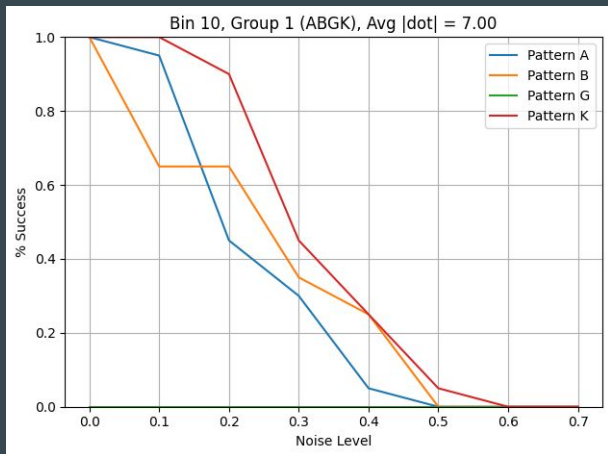
Este gráfico muestra el porcentaje promedio de aciertos en la recuperación de patrones según el nivel de ruido, para diferentes niveles de ortogonalidad (representados por los índices de los bins). Para cada bin se seleccionaron 2 grupos de patrones, y en cada nivel de ruido se calculó el promedio de aciertos sobre 20 iteraciones, considerando todos los patrones de ambos grupos.



Desempeño por patrón por grupo

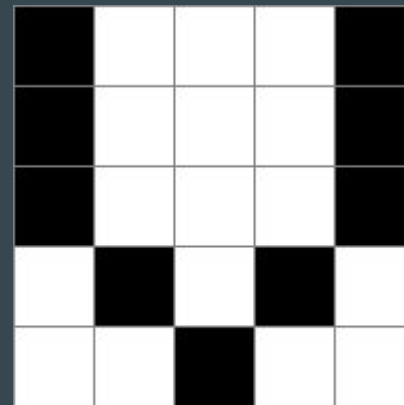
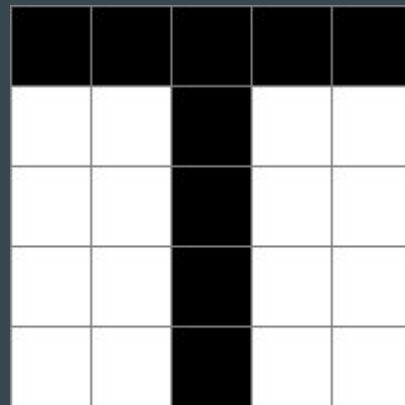
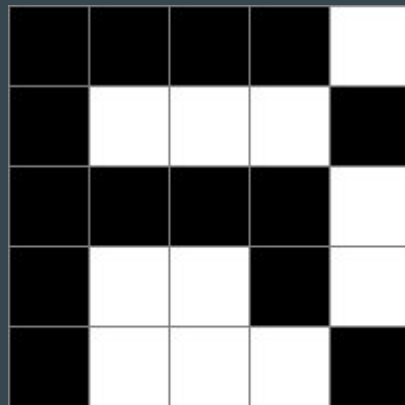
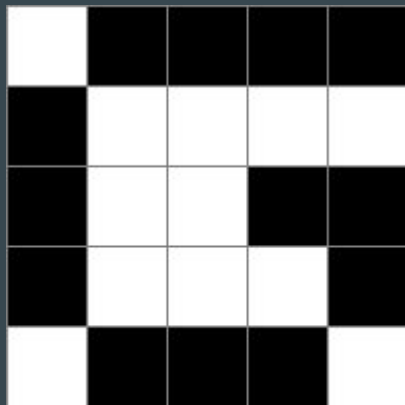


Desempeño por patrón por grupo

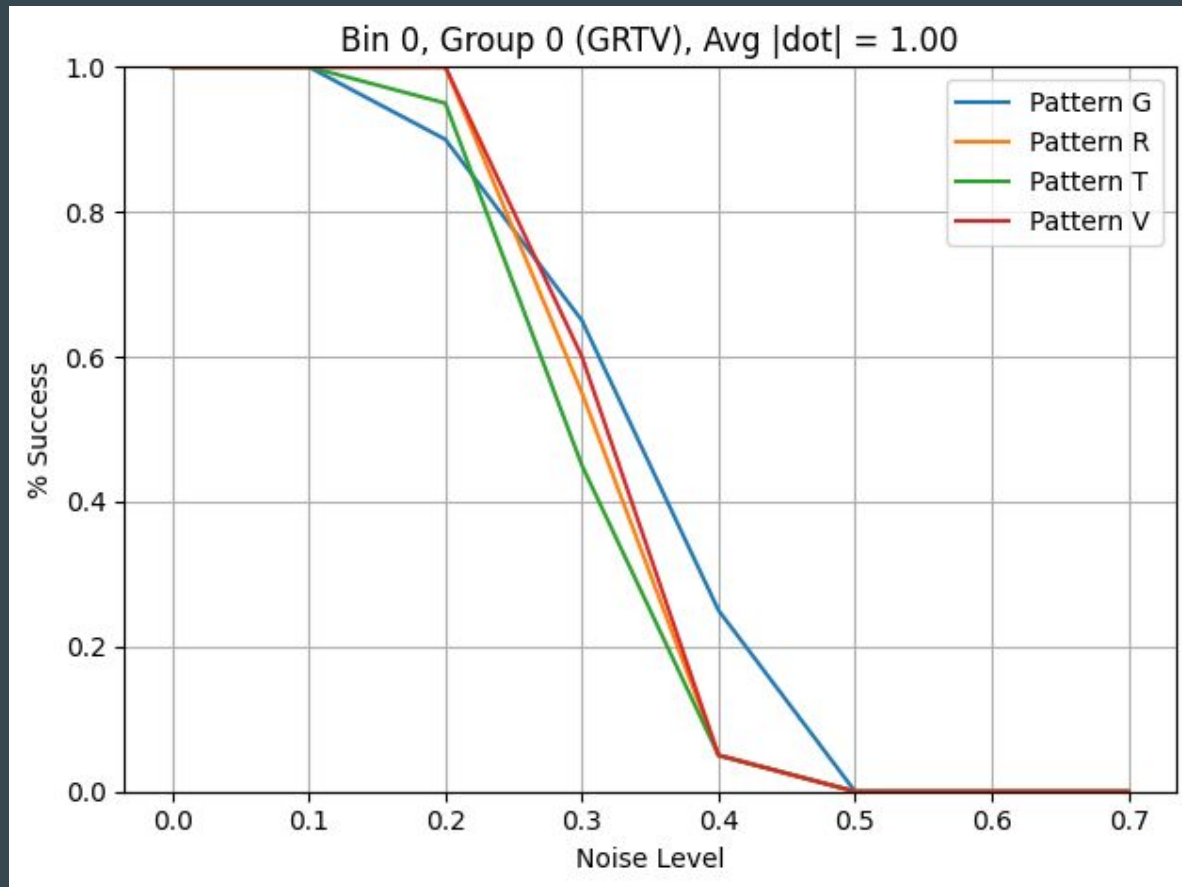


Patrones elegidos

Para las siguientes pruebas elegimos el grupo de patrones “G R T V”, con promedio de producto escalar de 1, el mínimo encontrado.

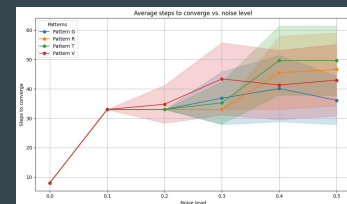
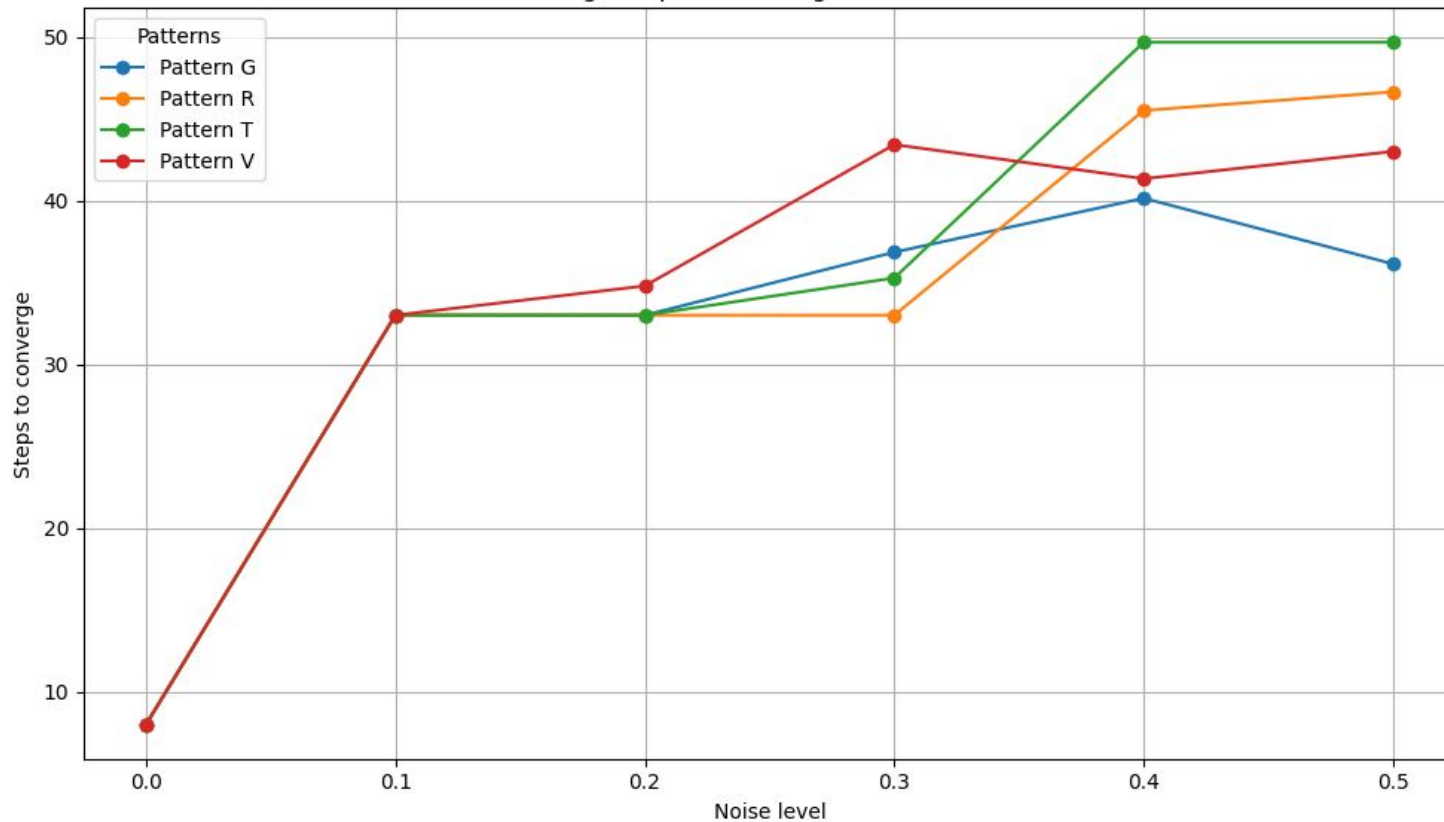


Desempeño por patrón por grupo

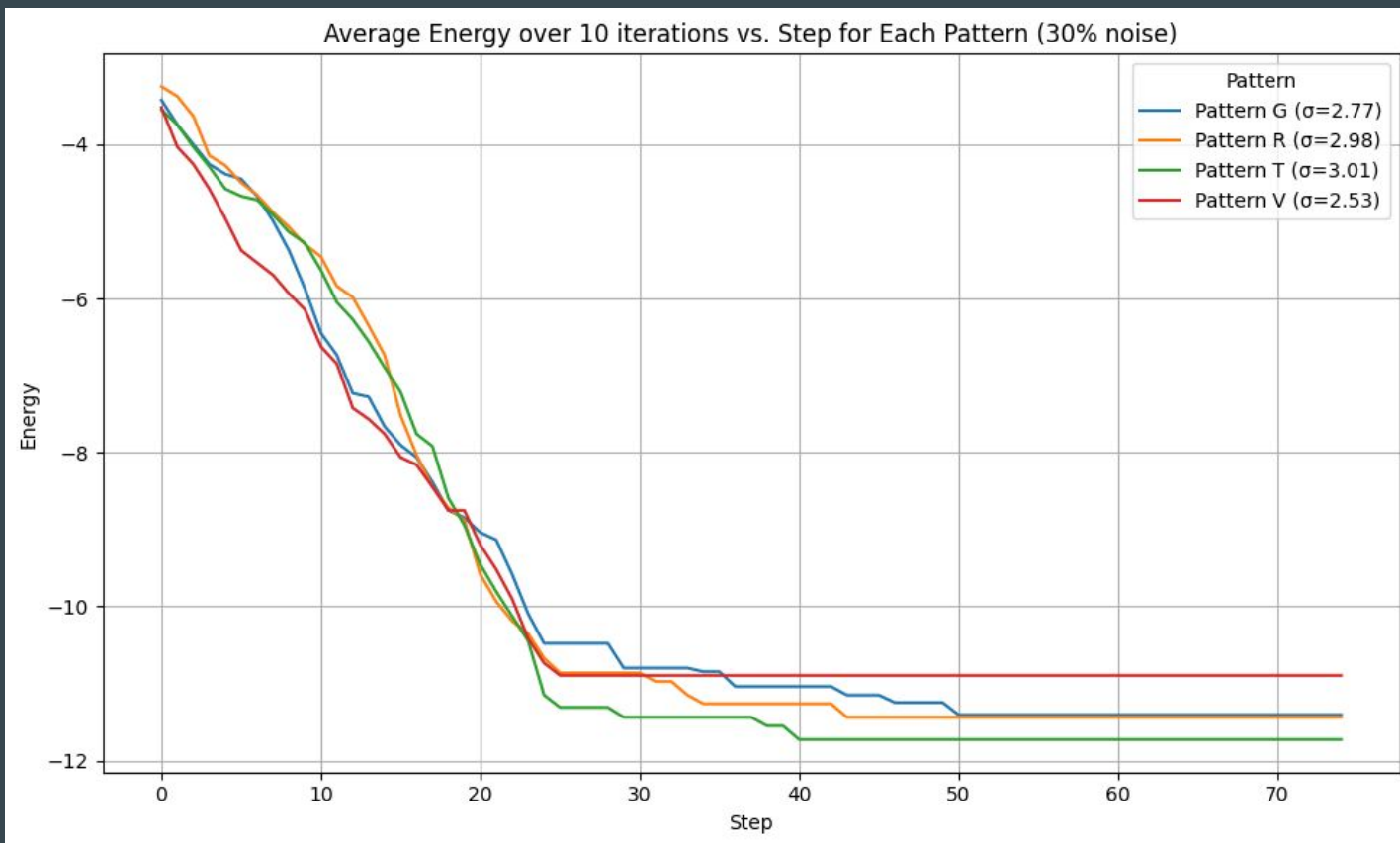


Convergencia

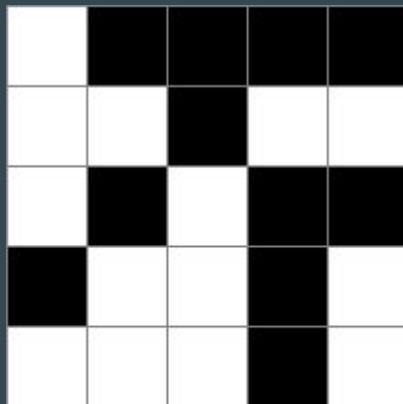
Average steps to converge vs. noise level



Energía



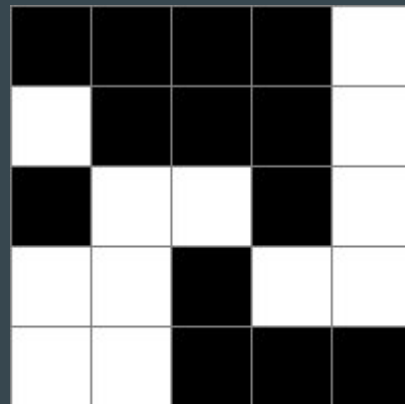
Convergencia



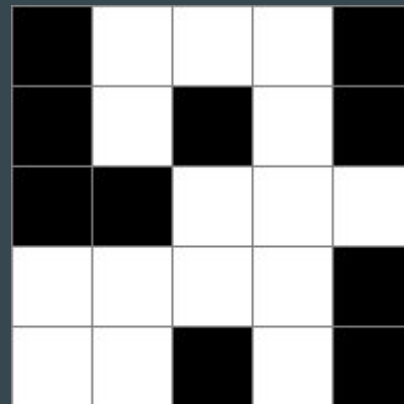
Step 1
Expected: G



Step 1
Expected: R



Step 1
Expected: T



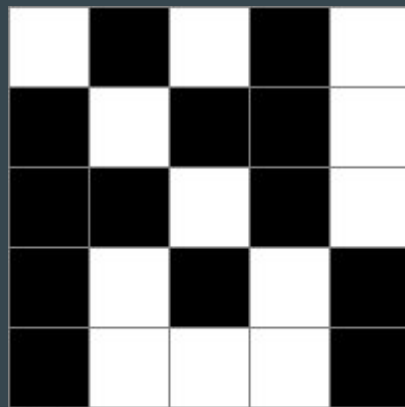
Step 1
Expected: V

Patrones espurios

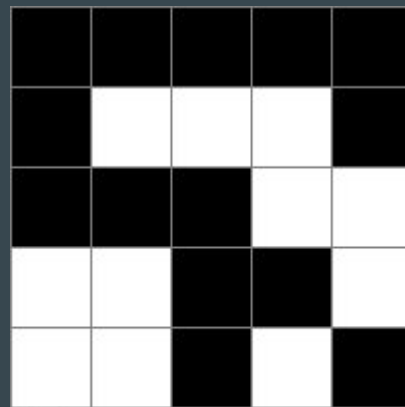
Acá se tiene una muestra de patrones espurios a los que convergió la red para cada patrón.



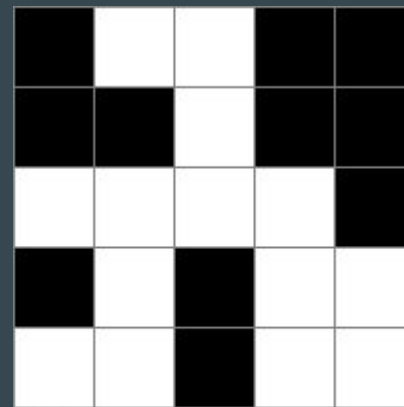
Step 1
Expected: G-Spur-Gen



Step 1
Expected: R



Step 1
Expected: T



Step 1
Expected: V

FIN

