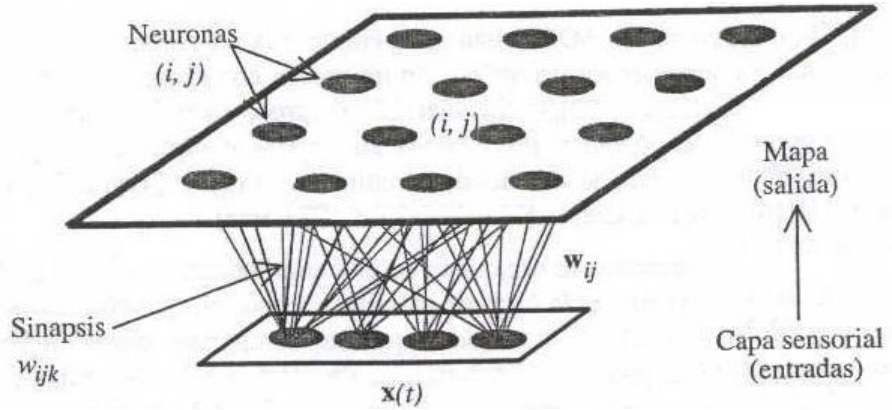
Sistemas de inteligencia artificial
TP4: Métodos de Aprendizaje NO Supervisado

Grupo 19
Integrantes:
- Lucas Catolino
- Matias Ricarte

Ejercicio 1.a

Redes de Kohonen

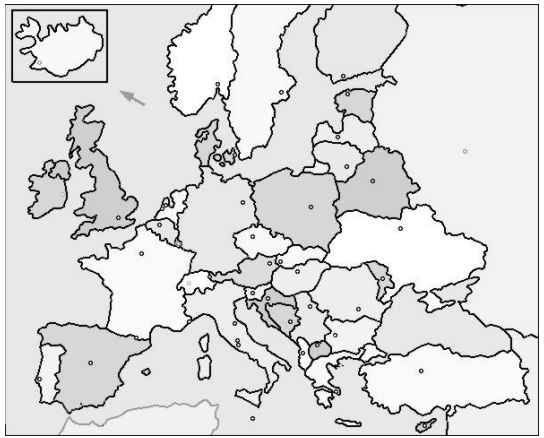
- Las neuronas se conectan consigo mismas positivamente, y con las vecinas negativamente
- Se activa una neurona (aprendizaje competitivo)
- Una sola capa de kxk neuronas
- Agrupan datos
- Limitaciones:
  - Si la entrada es muy grande es difícil bajar la dimensionalidad
  - Sólo para variables numéricas
  - No existe un criterio para elegir el k



Redes de Kohonen

El problema: dados datos geo-socio-económicos de 28 países europeos, agruparlos por características similares

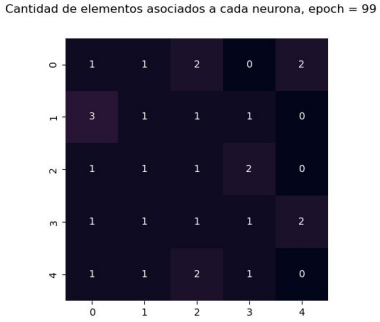
	Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0	Austria	83871	41600	3.5	79.91	0.8	0.03	4.2
1	Belgium	30528	37800	3.5	79.65	1.3	0.06	7.2
2	Bulgaria	110879	13800	4.2	73.84	2.6	-0.80	9.6



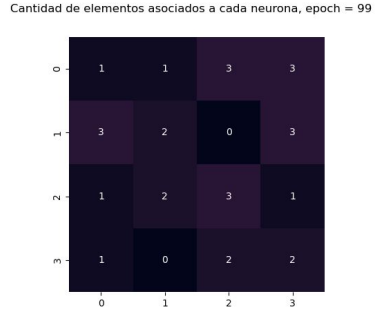
Experimentación

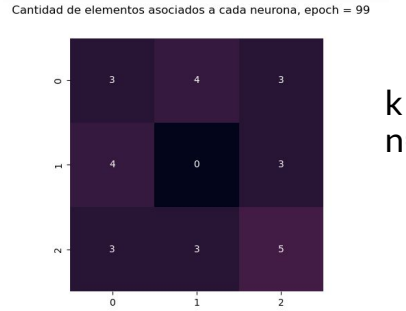
- Encontramos que usar una grilla de k = 3 no resultaba en neuronas muertas.
- Elegimos usar k = 3 para poder tener más clasificaciones para el conjunto de datos.
- Encontramos que algunos casos con k = 3 y una tasa de aprendizaje menor que 0.7 podía ocurrir que al final de las iteraciones existieran neuronas muertas.

k = 5



k = 4



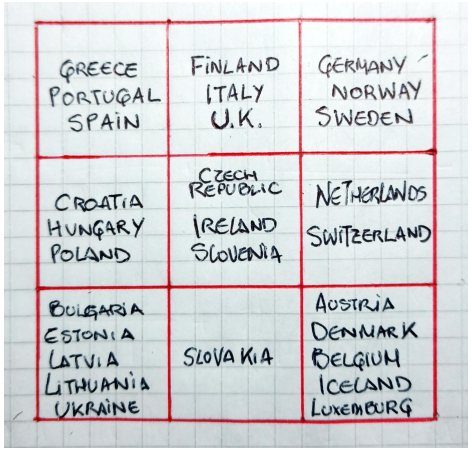


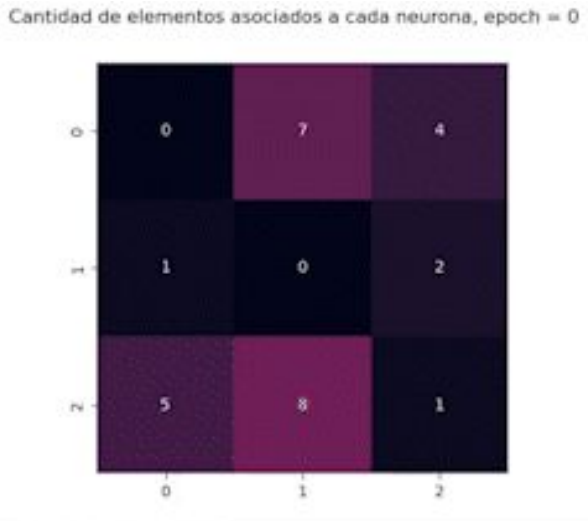
k = 3
n=0.3

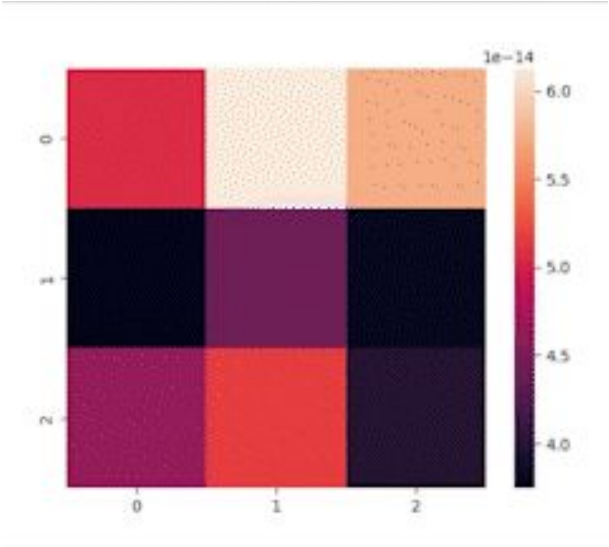
Resultados

Parámetros usados:

- k=3
- learning rate = 0.7
- R = 10



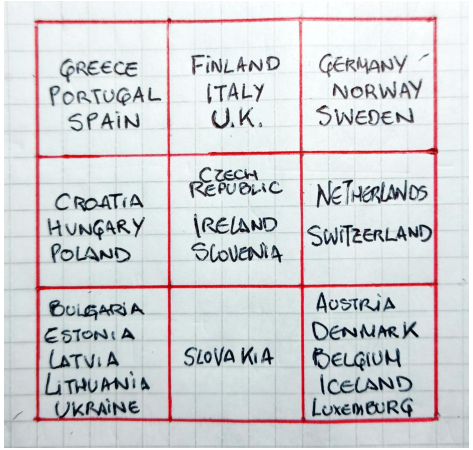




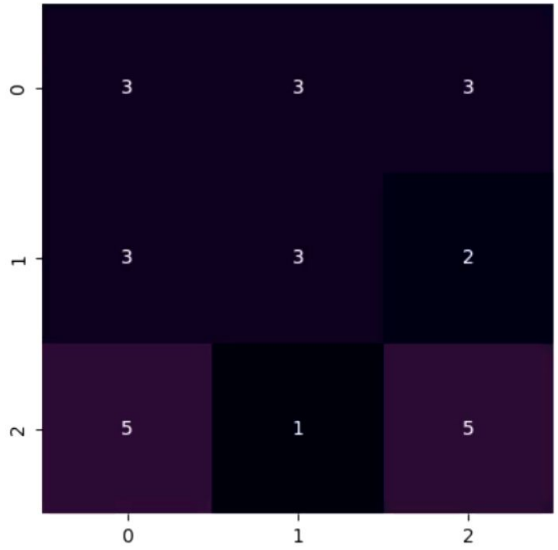
Resultados finales

Parámetros usados:

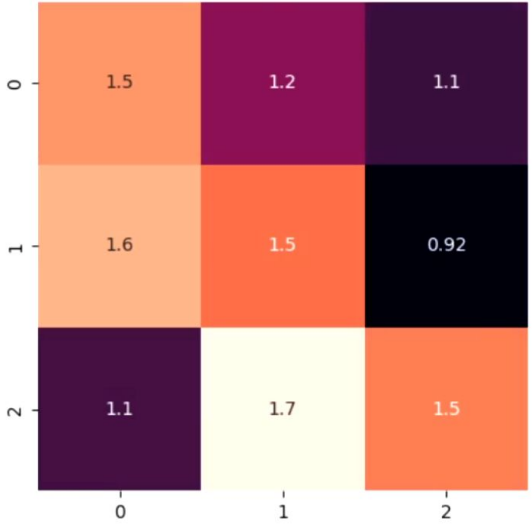
- k=3
- learning rate = 0.7
- R = 10



Cantidad de elementos asociados a cada neurona, epoch = 99



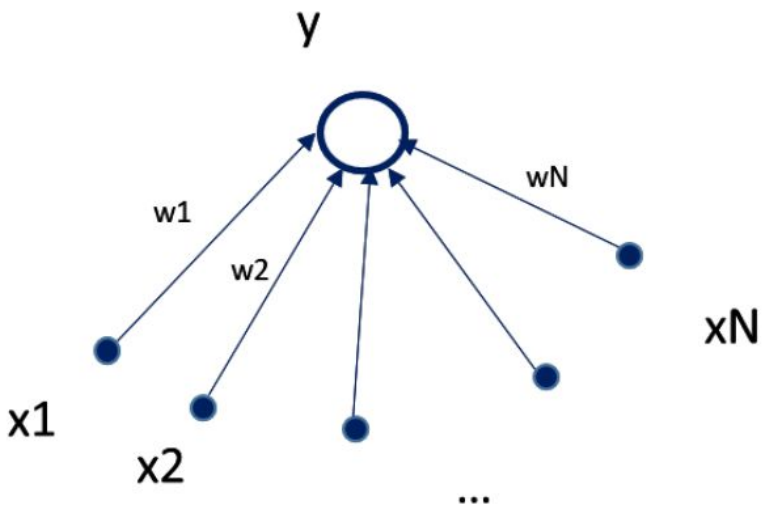
Distancia promedio entre neuronas vecinas, epoch = 99



Ejercicio 1.b

Regla de Oja

- Perceptrón lineal simple
- Busca construir la primera componente principal (autovector asociado al mayor autovalor de la matriz de correlaciones)

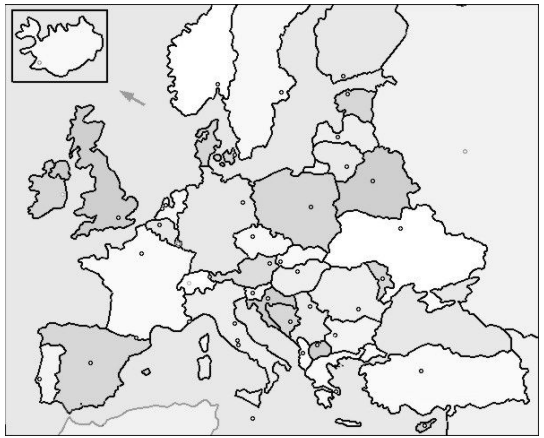


$$y^n = \sum_{j=1}^{N} w_j^n * x_j^i$$

Regla de Oja

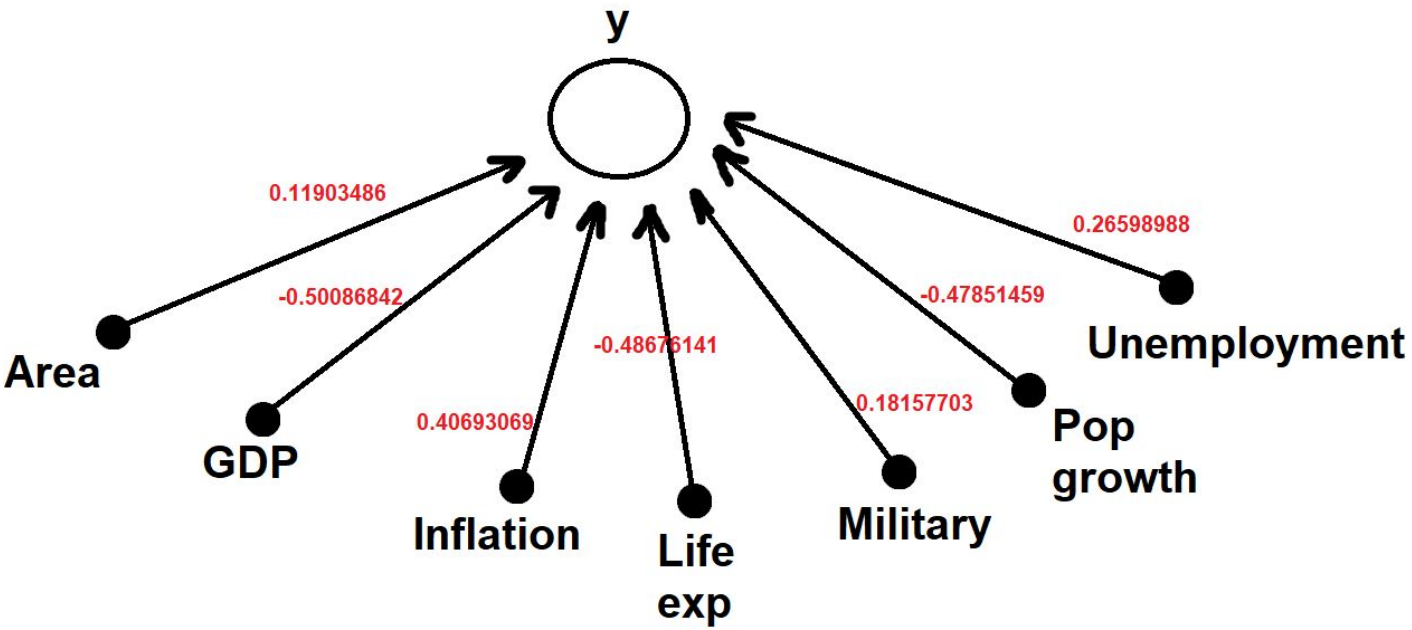
El problema: dados datos geo-socio-económicos de 28 países europeos, calcular la primera componente principal

	Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0	Austria	83871	41600	3.5	79.91	0.8	0.03	4.2
1	Belgium	30528	37800	3.5	79.65	1.3	0.06	7.2
2	Bulgaria	110879	13800	4.2	73.84	2.6	-0.80	9.6



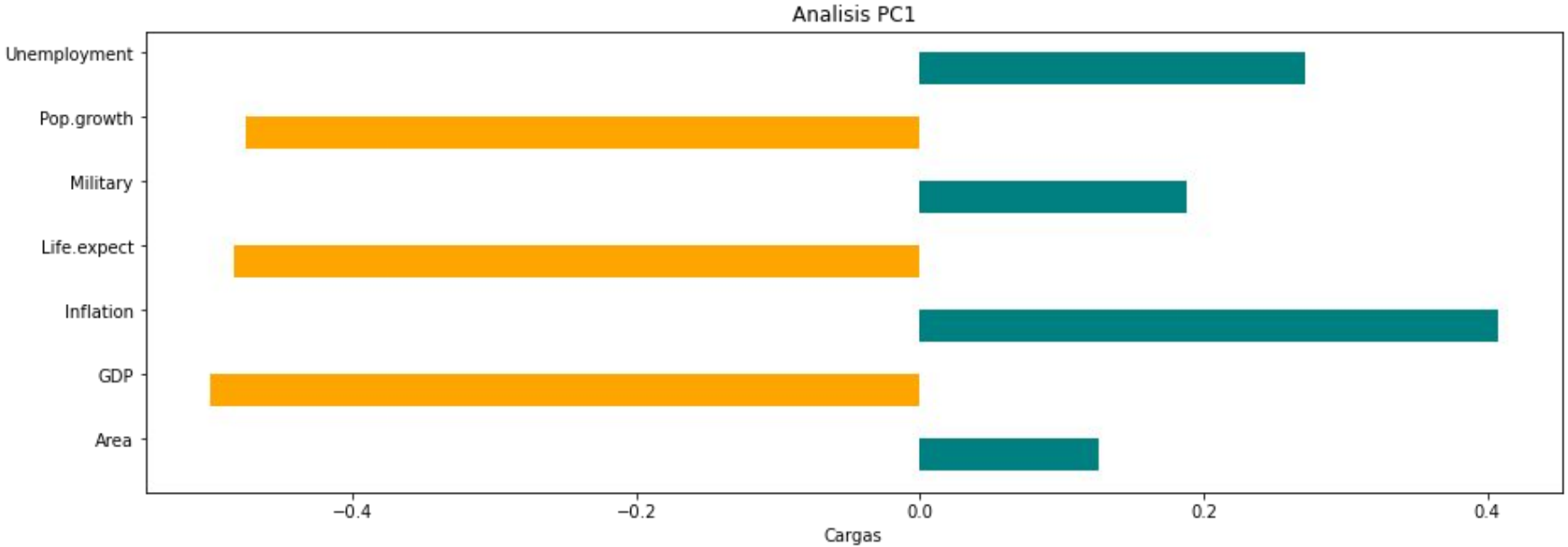
Regla de Oja

Epochs: 5000
Learning rate: 0.0001

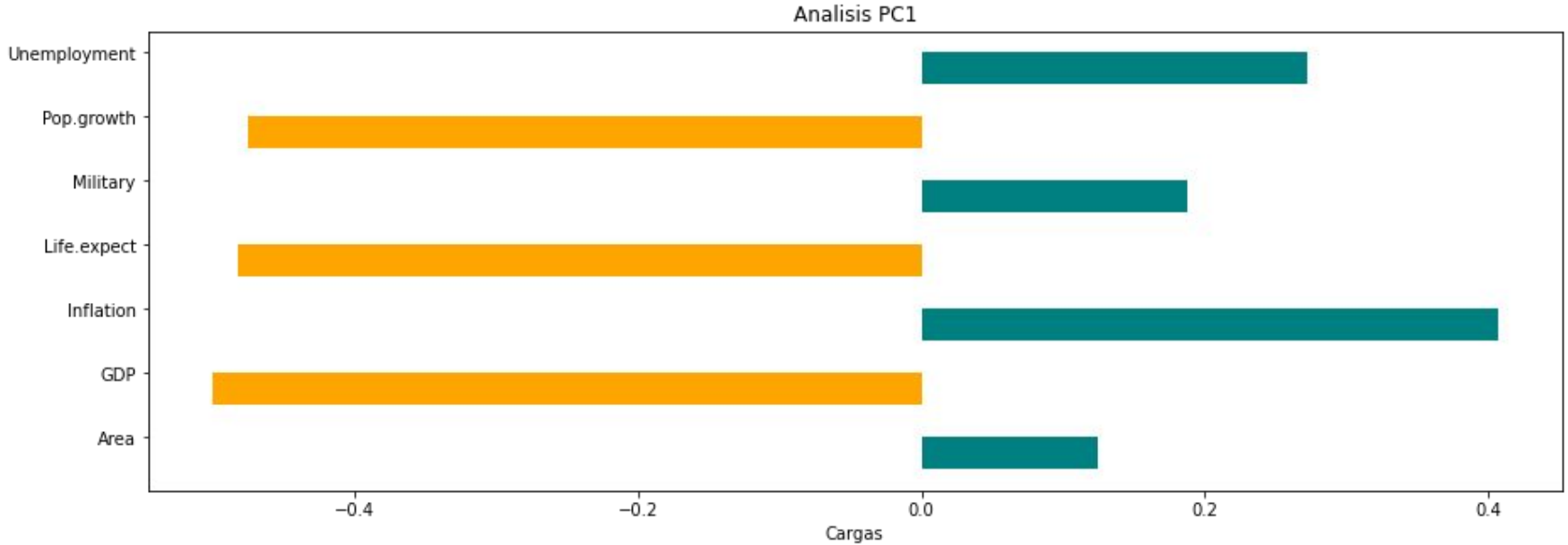


Regla de Oja

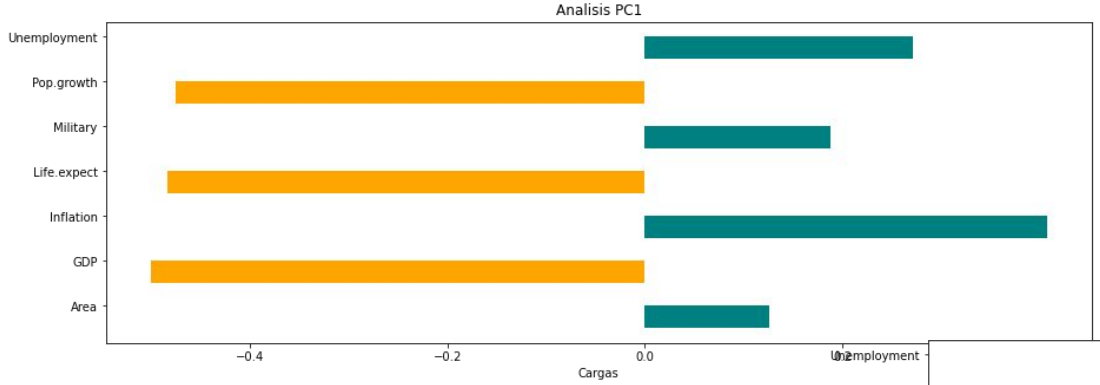
Epochs: 5000
Learning rate: 0.0001

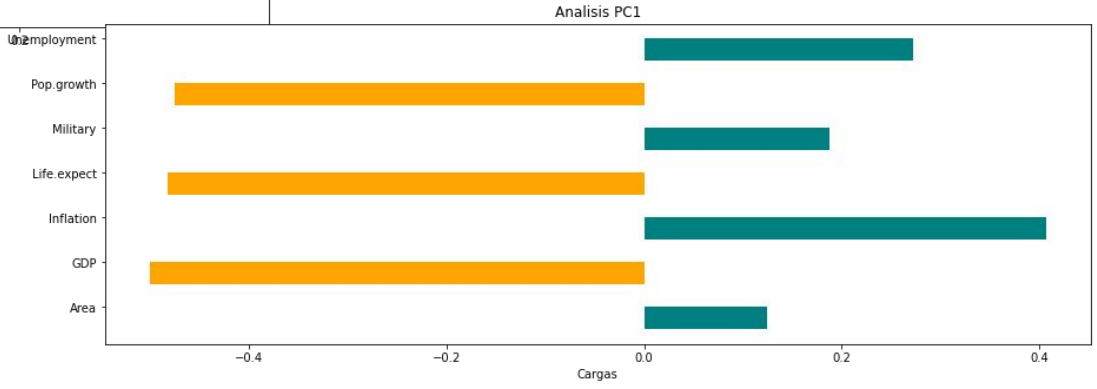


sklearn (PCA)


Analisis PC1
Unemployment
Pop.growth
Military
Life.expect
Inflation
GDP
Area
−0.4
−0.2
0.0
0.2
0.4
Cargas

Regla de Oja vs sklearn (PCA)



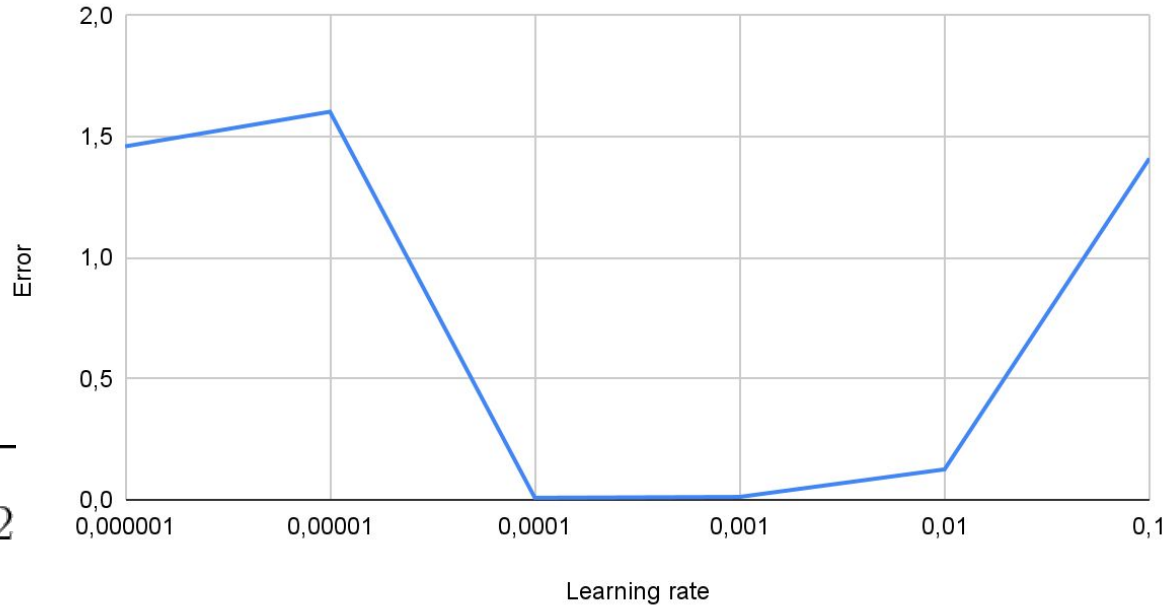


Regla de Oja

Learning rate	Error
0,000001	1,460024964
0,00001	1,603104589
0,0001	0,008344459362
0,001	0,0120035223
0,01	0,1256710671
0,1	1,410198854

Epochs= 1000

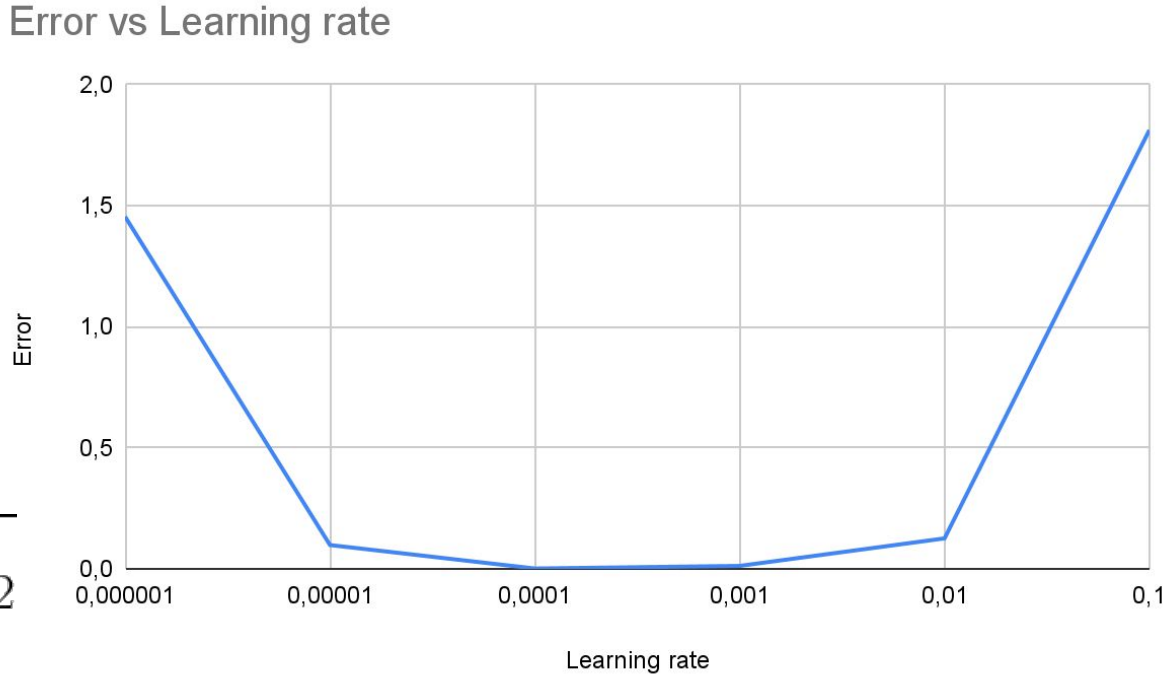
$$Error = \sqrt{\sum_{i=1}^{7} \left(W_i - W_i^L\right)^2}$$



Regla de Oja

Learning rate	Error
0,000001	1,45466362
0,00001	0,09815249681
0,0001	0,00119472466
0,001	0,0120035223
0,01	0,1256710671
0,1	1,812459308

Epochs= 5000



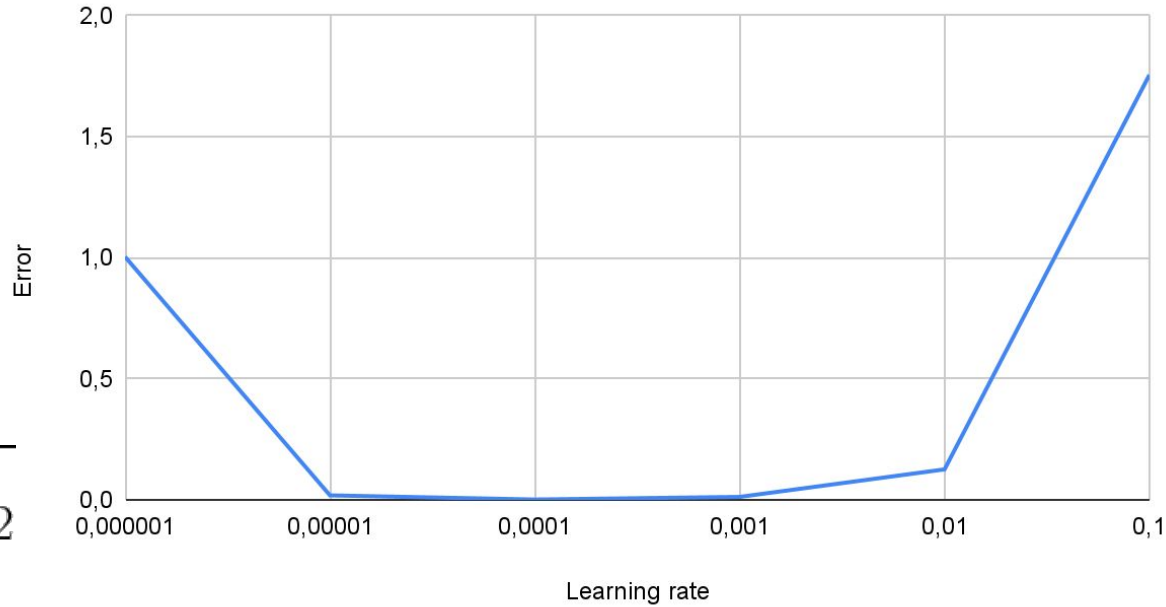
$$Error = \sqrt{\sum_{i=1}^{7} \left(W_i - W_i^L\right)^2}$$

Regla de Oja

Learning rate	Error
0,000001	1,003595871
0,00001	0,01828034419
0,0001	0,001194724648
0,001	0,0120035223
0,01	0,1256710671
0,1	1,755334835

Epochs= 10000

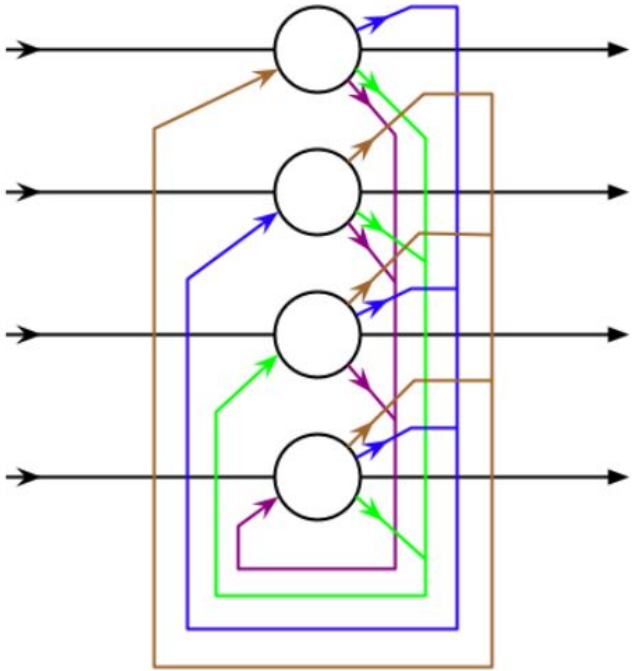
$$Error = \sqrt{\sum_{i=1}^{7} \left(W_i - W_i^L\right)^2}$$



Ejercicio 2

Redes de Hopfield

- Todas las neuronas conectadas con las otras (menos consigo mismas)
- Salida binaria (activo o no)
- Una sola capa
- Llevan a un patrón
- Limitaciones:
  - Guarda hasta el 15% de la entrada
  - Patrones de referencia aproximadamente ortogonales



Redes de Hopfield

El problema: almacenar 4 patrones de letras de 5x5 (1 o -1), y a partir de una entrada con ruido asociar a un patrón

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

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

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

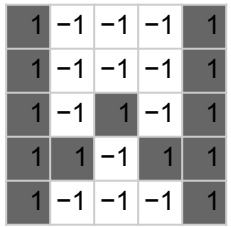
1	−1	−1	−1	1
−1	1	−1	1	−1
−1	−1	1	−1	−1
−1	1	−1	1	−1
1	−1	−1	−1	1

Ortogonalidad:

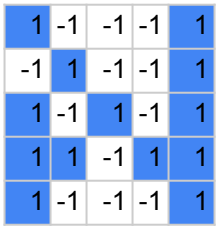
- L, X: 1
- L, T: -3
- L, P: -1
- X, T: 1
- X, P: -5
- T, P: 3

Redes de Hopfield

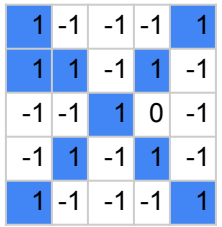
Ruido: 0.2



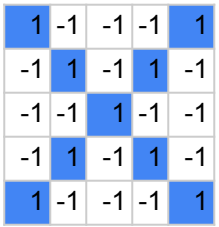
Entrada sin ruido



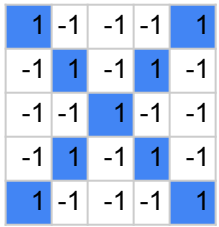
Entrada con ruido



Época 1



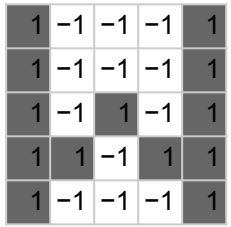
Época 2



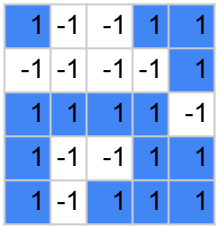
Época 3: patrón estabilizado

Redes de Hopfield

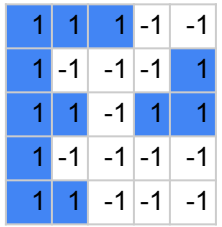
Ruido: 0.4



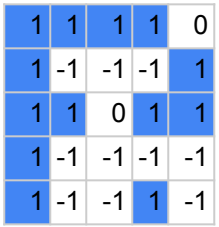
Entrada sin ruido



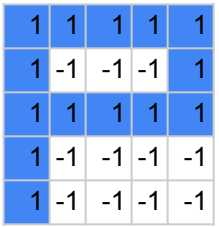
Entrada con ruido



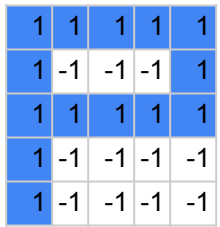
Época 1



Época 2



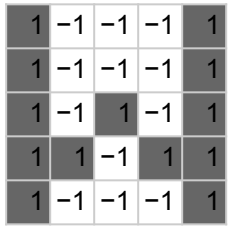
Época 3



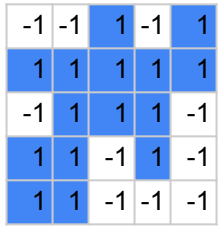
Época 4: patrón estabilizado

Redes de Hopfield

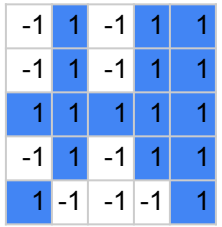
Ruido: 0.6



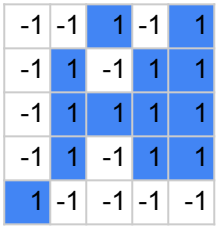
Entrada sin ruido



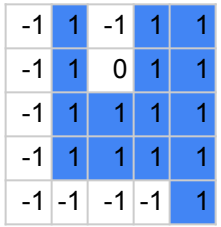
Entrada con ruido



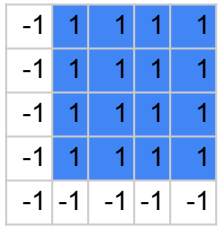
Época 1

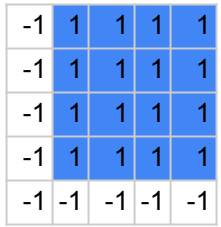


Época 2



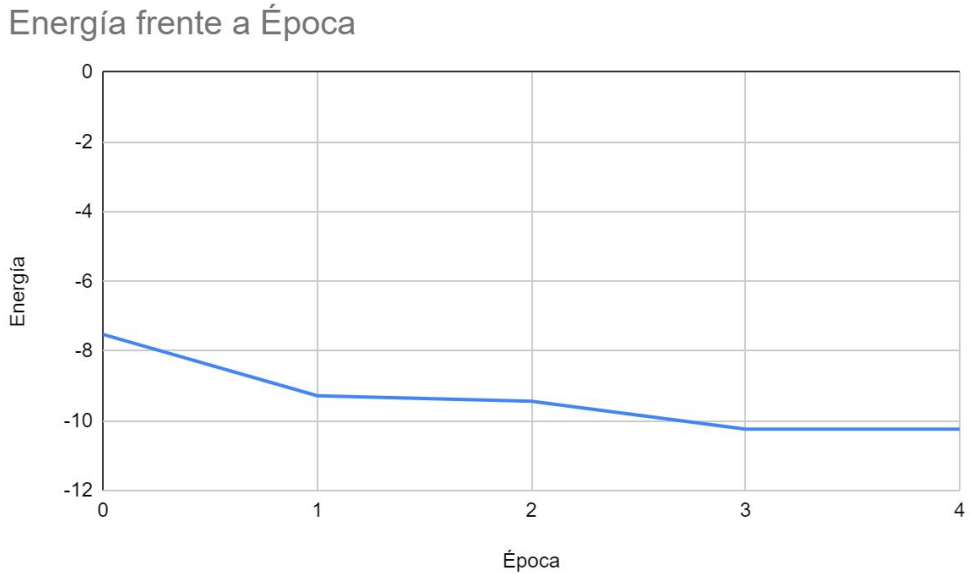
Época 3





Época 5: estado espúreo

Redes de Hopfield



Patrón entrante: T con ruido 0.3

Hopfield demostró que la red converge ya que la energía decrece, y los mínimos locales son patrones

Un estado espúreo es un mínimo (local) en la función de energía

$$H = -\sum_{j>i} w_{ij} S_i S_j$$

Muchas gracias

Grupo 19
Integrantes:

- ➔ Lucas Catolino
- ➔ Matias Ricarte