Resolución TP Programación 1

Clasificador basado en Ley de Aprendizaje Competitivo

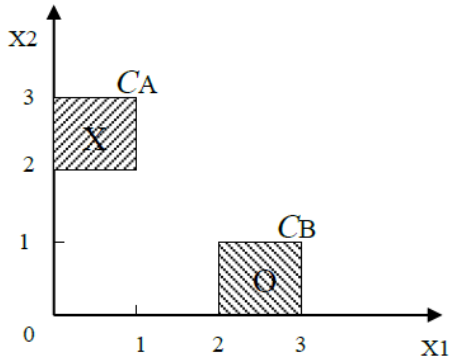
Alumno: Salomón Nicolás

Curso: Inteligencia Artificial aplicada a la Identificación y Control

Profesor: Dr. Ing. H. Daniel Patiño

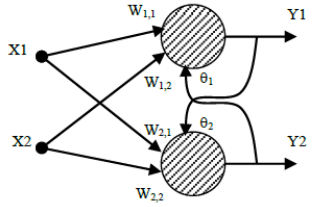
Año: 2022

1. **Desarrollar el código de un algoritmo que realice la tarea de *Clasificación* basado en una ley de *aprendizaje competitivo* capaz de clasificar los *Clusters A y B*, considerando un conjunto de datos de entrada para el entrenamiento infinito. Las Clases se muestran en la siguiente figura:**



Proposiciones:

1. Dado el condicionamiento de emplear una ley de aprendizaje competitivo para la clasificación de dos clases, se propone la siguiente estructura de la red neuronal $RN_{2,2}$:



2. Se proponen dos leyes de aprendizaje no supervisadas competitivas, con y sin señales laterales inhibitorias:

- *Si la neurona 1 "gana" sus sinapsis se ajustan según la siguiente ley de aprendizaje:*

$$w_{1i}(k+1) = w_{1i}(k) + \Delta w_{1i}$$

$$\Delta_{1i}\begin{cases} \eta(x_i - w_{1i}), & si\ nuerona\ 1\ "gana" \\ 0 & si\ \ neurona\ 1\ "pierde" \end{cases}$$

- *Ley de Aprendizaje Competitivo con señales laterales inhibitorias:*

$$\Delta_{\theta_1} = \alpha\ \bar{y}_2\ \theta_{12} \qquad si\ \ 1\ "gana"$$
$$\Delta_{\theta_1} = -\alpha\ y_2\ \theta_{12} \qquad si\ \ 1\ "pierde"$$

a) Generar el código para la realización de la simulación computacional.

b) Establecer un criterio para detener el proceso de aprendizaje.

c) Dibujar los vectores de peso aprendidos.

d) Determinar el % de acierto para una muestra significativa.

El código de desarrollo del presente trabajo se encuentra adjunto al mismo (TP_Programación_1 - Aprendizaje_Competitivo.ipynb)

Para desarrollar el presente ejercicio se generaron puntos de forma aleatoria (en orden aleatorio) pertenecientes a los dos clusters posibles, según se observa en la Figura 1.

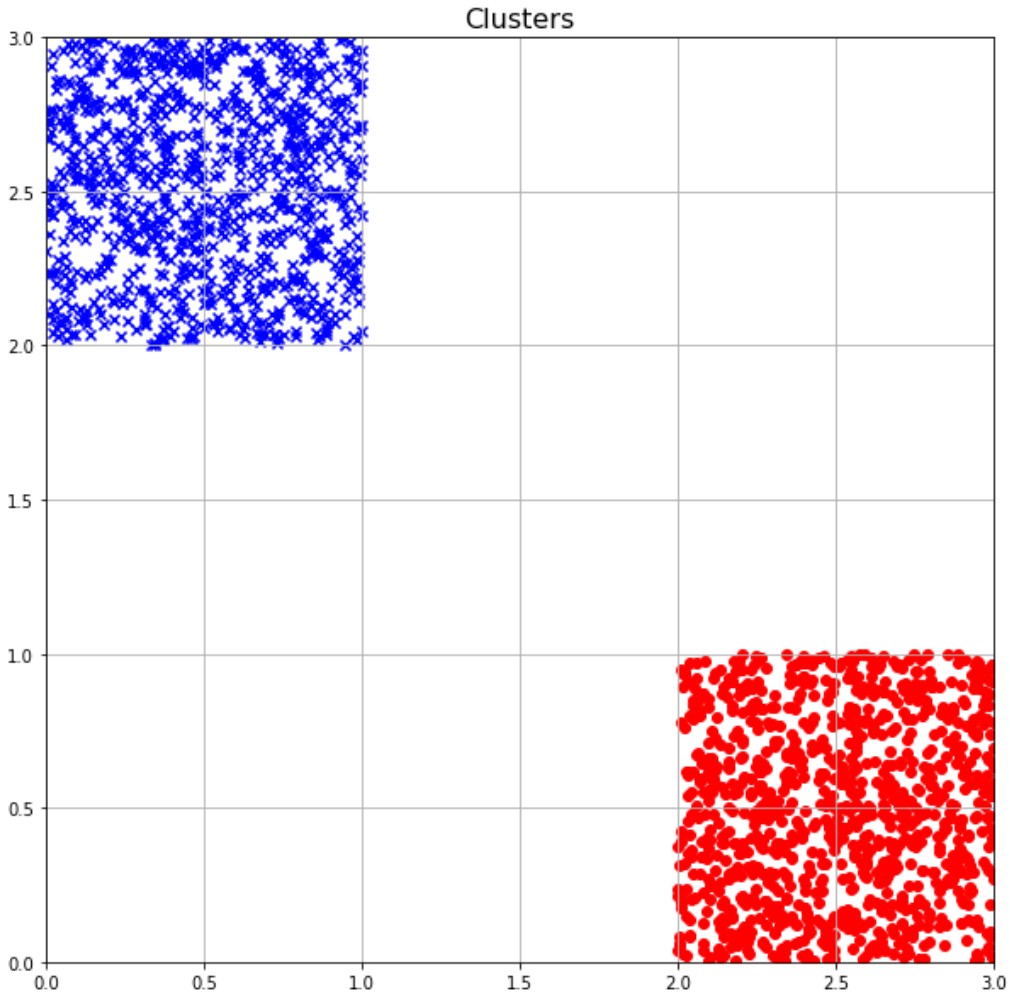


Figura 1. Visualización de los clusters.

Luego de programar el algoritmo generador de puntos se procedió a generar el código necesario para implementar una red de 2 neuronas, como se muestra en el enunciado, siguiendo la primera propuesta de Ley de Aprendizaje, donde los pesos se actualizan de acuerdo a si la neurona "gana" o "pierde" su sinapsis.

Por último, se calculó el error cuadrático de las salidas de las neuronas como mse = sqrt(Δw1^2 + Δw2^2). En base al error calculado y a un umbral fijado de manera empírica se detiene el proceso de entrenamiento y se muestra por pantalla el número de la última iteración junto con los valores de pesos iniciales y finales.

El resultado del entrenamiento puede apreciarse en la Figura 2 (variación del valor del vector de pesos), donde se alcanzó un mse = 0.0087 aproximadamente, con la iteración: 2339.

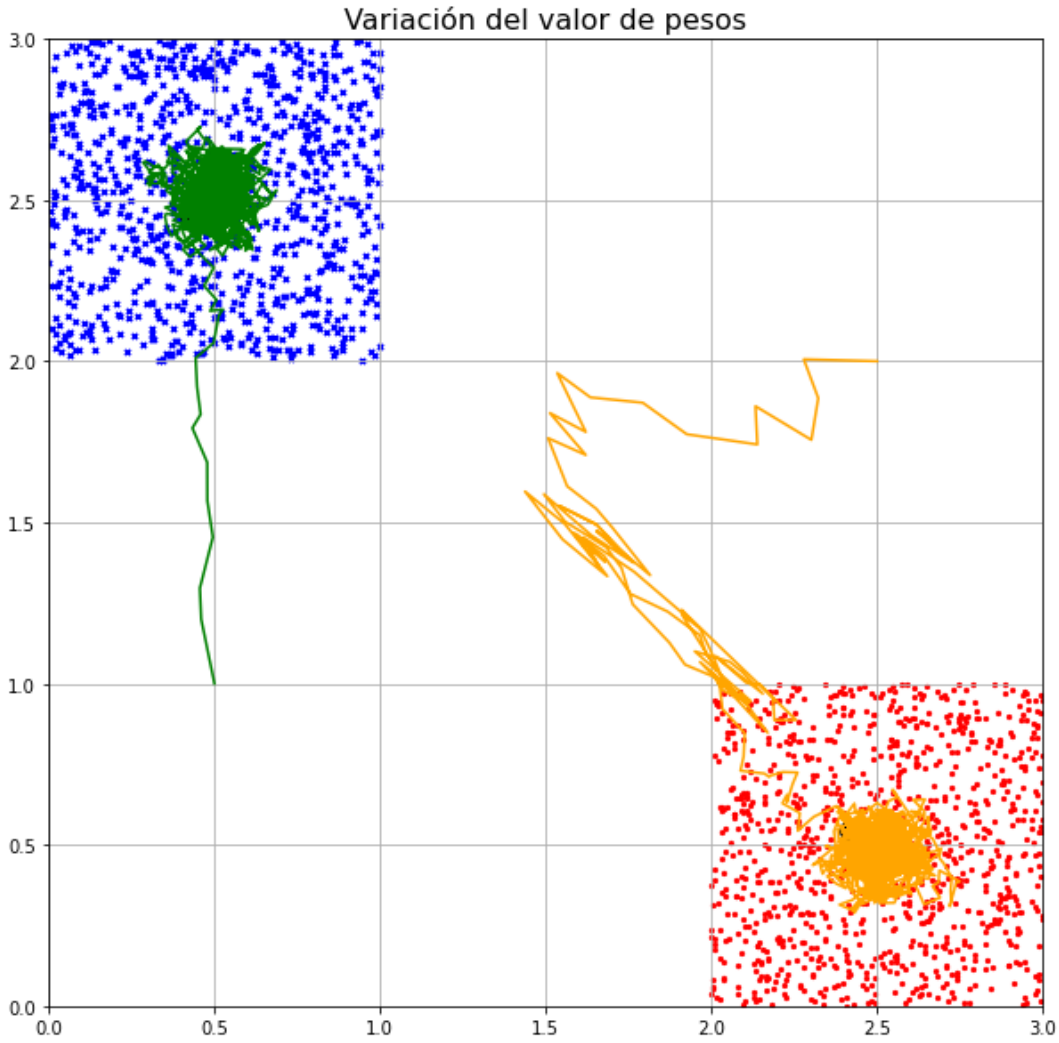


Figura 2. Resultado del entrenamiento de la red neuronal.

Por último, se muestra en la Figura 3 la región de decisión generada por nuestra red neuronal ya entrenada sobre un dataset de prueba generado.

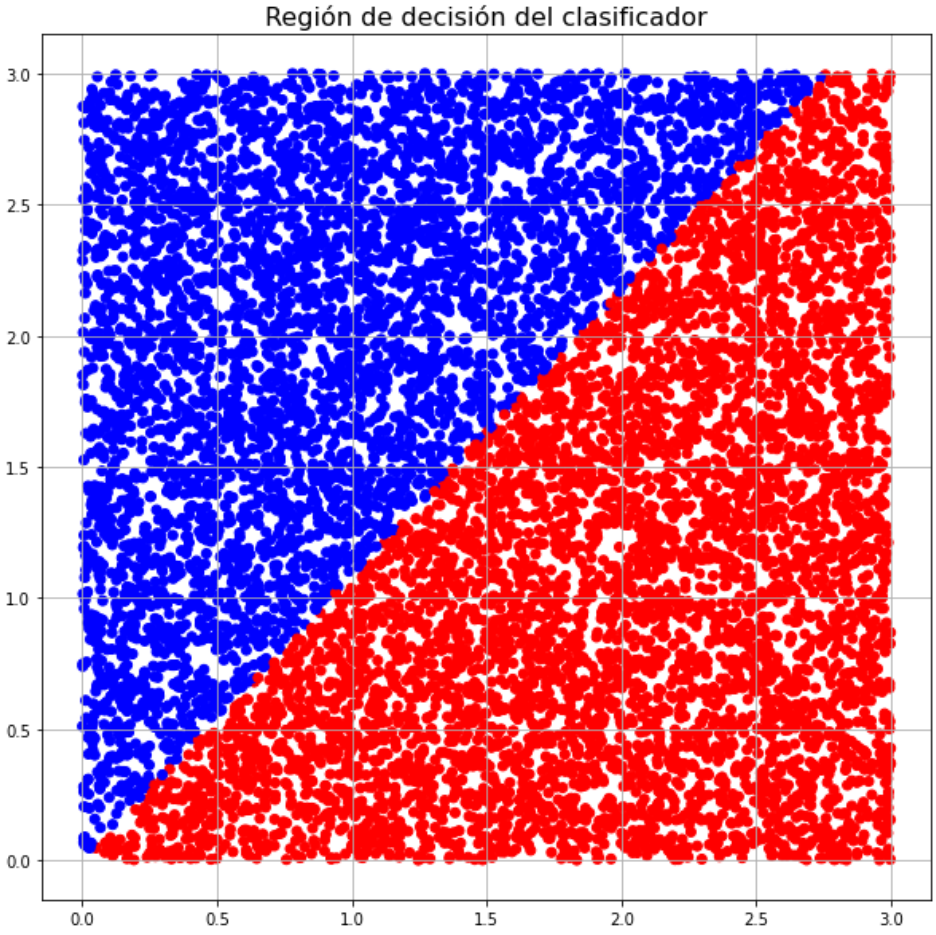


Figura 3. Región de decisión generada.

Se puede observar cómo se traza una región casi perfecta en donde todos los puntos rojos están más cerca del centro del cluster B (2.5,0,5) que del centro del cluster A (0.5,2.5). Análogamente, todos los puntos azules están más cerca del centro del cluster A (0.5,2.5) que del centro del cluster B (2.5,0,5).