

UNIVERSIDAD DE ORIENTE  
NÚCLEO ANZOÁTEGUI  
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACION  
CIRCUITO Y SISTEMAS



**Hopfield.**

Profesor:

Jose Bastardo

Nombre:

Eulises Brazon

Barcelona, 6 de Agosto del 2023

## Descripción del problema

El problema al que se dio solución en este proyecto es el de implementar una red de Hopfield y una interfaz gráfica para la recuperación de patrones, la red de Hopfield es una red neuronal recurrente utilizada para almacenar y recuperar patrones a partir de estímulos parciales o ruidosos.

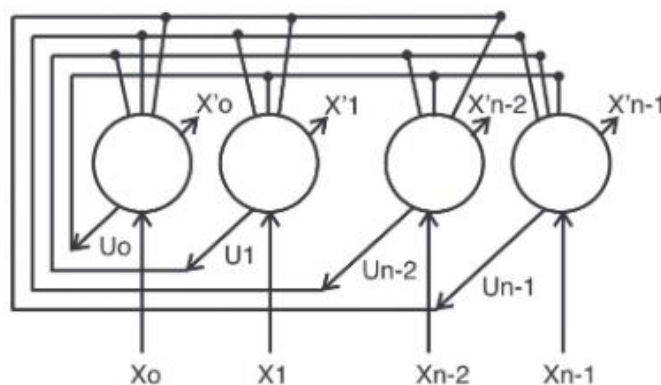
El objetivo principal era desarrollar un programa en Python que permitiera entrenar la red de Hopfield con un conjunto de patrones de entrada, normalizar los pesos de las conexiones sinápticas y luego utilizar la red para recuperar patrones basados en patrones de prueba proporcionados por el usuario.

Para lograr esto, se implementó la clase HopfieldNetwork que representa la red de Hopfield y contiene los métodos para entrenar y recuperar patrones, la matriz de pesos fue utilizada para almacenar las conexiones sinápticas entre las neuronas de la red. Además, se creó la clase HopfieldInterface que proporciona una interfaz gráfica para que el usuario pueda interactuar con la red, la interfaz permite ingresar patrones de prueba marcando y desmarcando celdas en una cuadrícula y luego realizar consultas a la red para recuperar los patrones correspondientes.

## Descripción de la solución

### Topología:

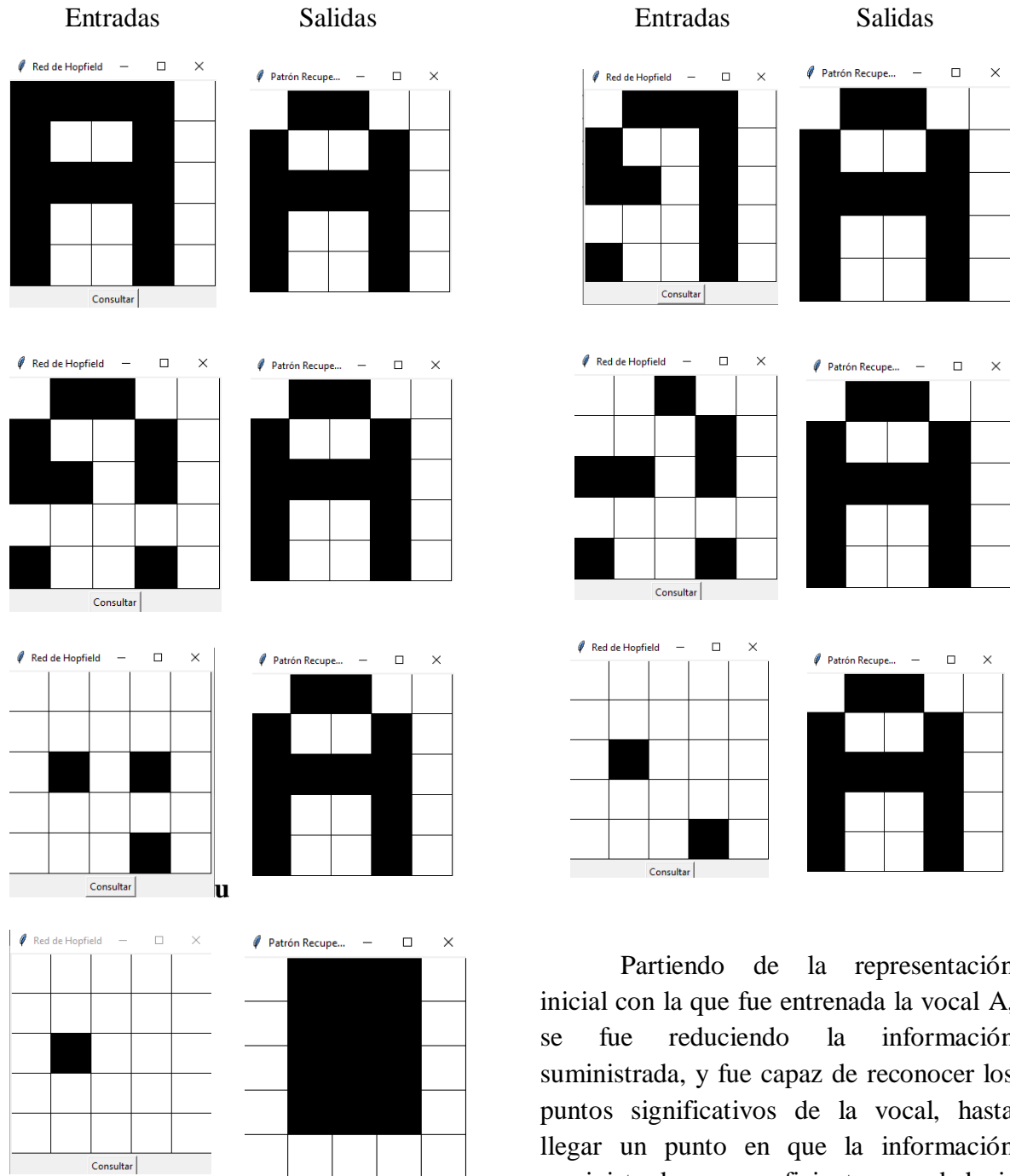
La red de Hopfield es una red neuronal recurrente con una topología completamente conectada, lo que significa que cada neurona está conectada a todas las demás neuronas en la red, es una red de una sola capa, sin capas ocultas.



(Modelo básico de la red Hopfield)

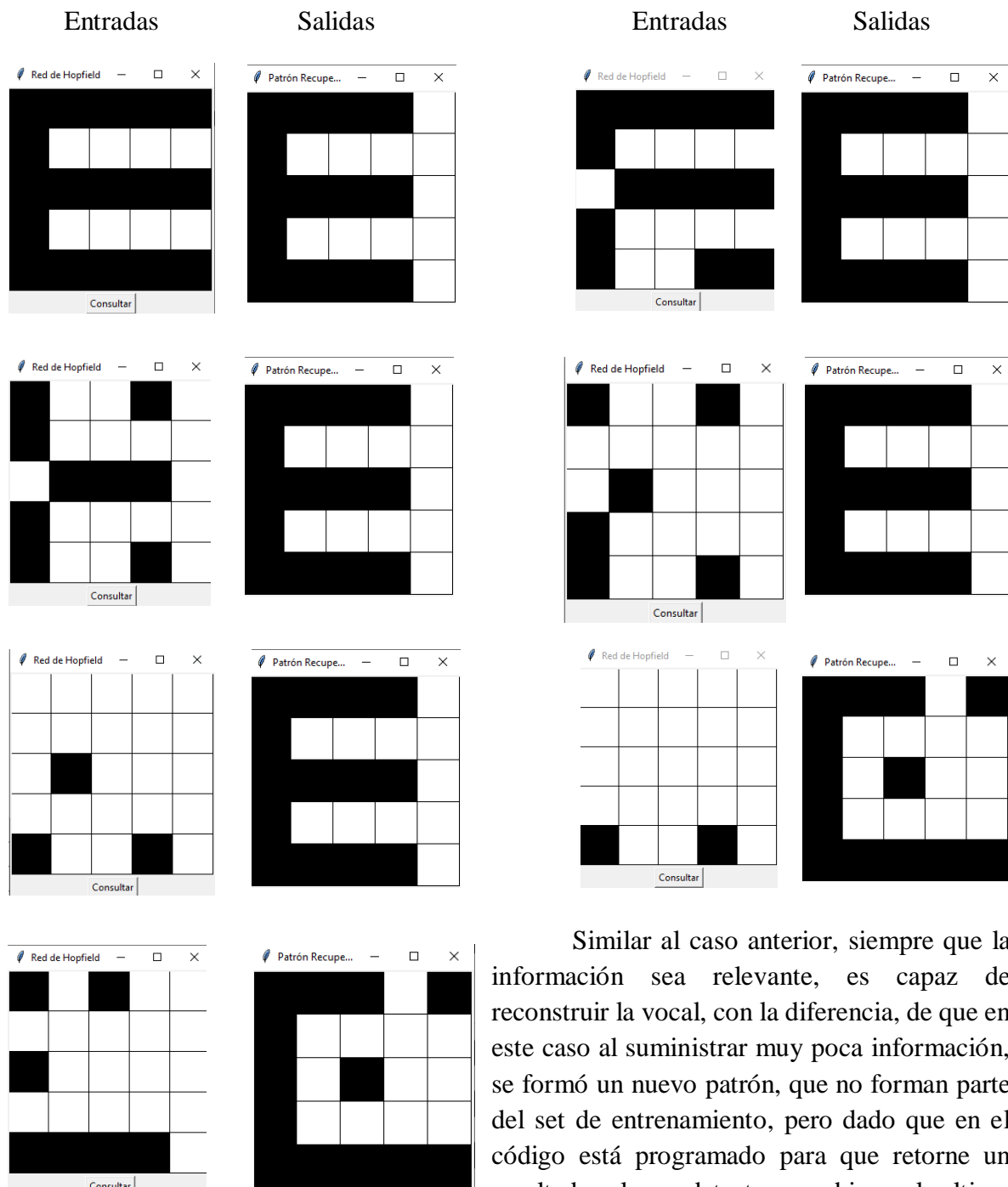
## Experimentos realizados

Pruebas con la vocal A:



Partiendo de la representación inicial con la que fue entrenada la vocal A, se fue reduciendo la información suministrada, y fue capaz de reconocer los puntos significativos de la vocal, hasta llegar un punto en que la información suministrada no es suficiente para deducir un resultado adecuado.

## Pruebas con la vocal E:

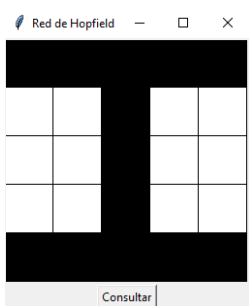


Similar al caso anterior, siempre que la información sea relevante, es capaz de reconstruir la vocal, con la diferencia, de que en este caso al suministrar muy poca información, se formó un nuevo patrón, que no forman parte del set de entrenamiento, pero dado que en el código está programado para que retorne un resultado al no detectar cambios, el ultimo

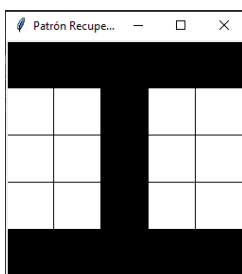
resultado fue la mejor adaptación que se logró conseguir dada la información suministrada, y los pesos con los que se encuentra configurado la red, usando ese nuevo patrón distinto, se hicieron varias pruebas y se observó que el patrón extraño, se puede formar con distintas entradas.

## Pruebas con la vocal I:

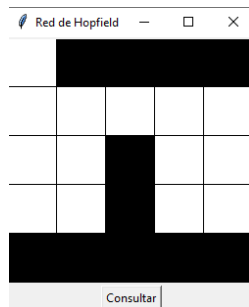
Entradas



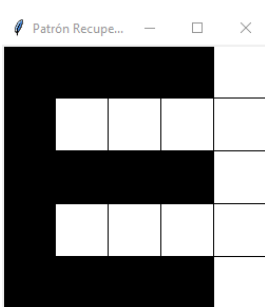
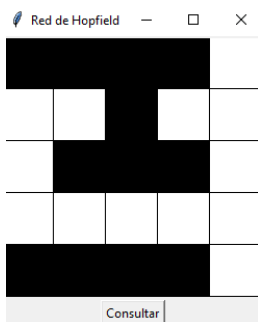
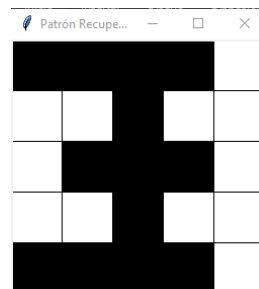
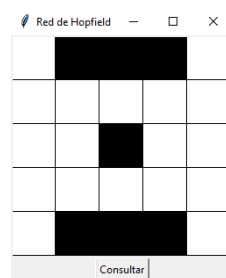
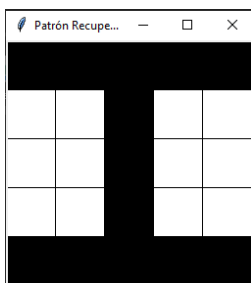
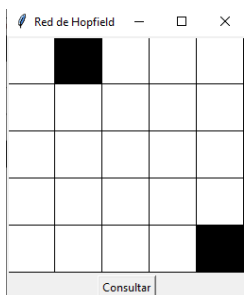
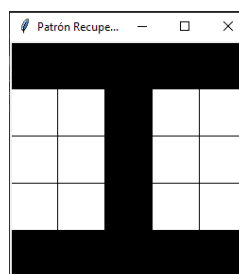
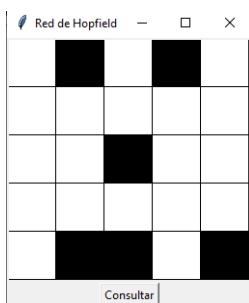
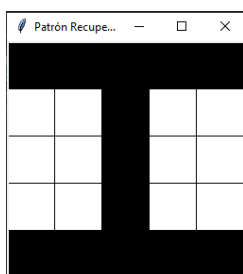
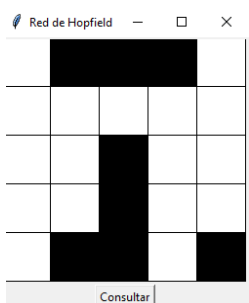
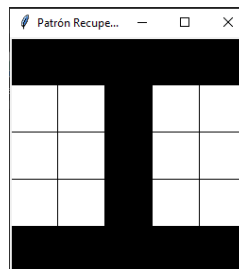
Salidas



Entradas

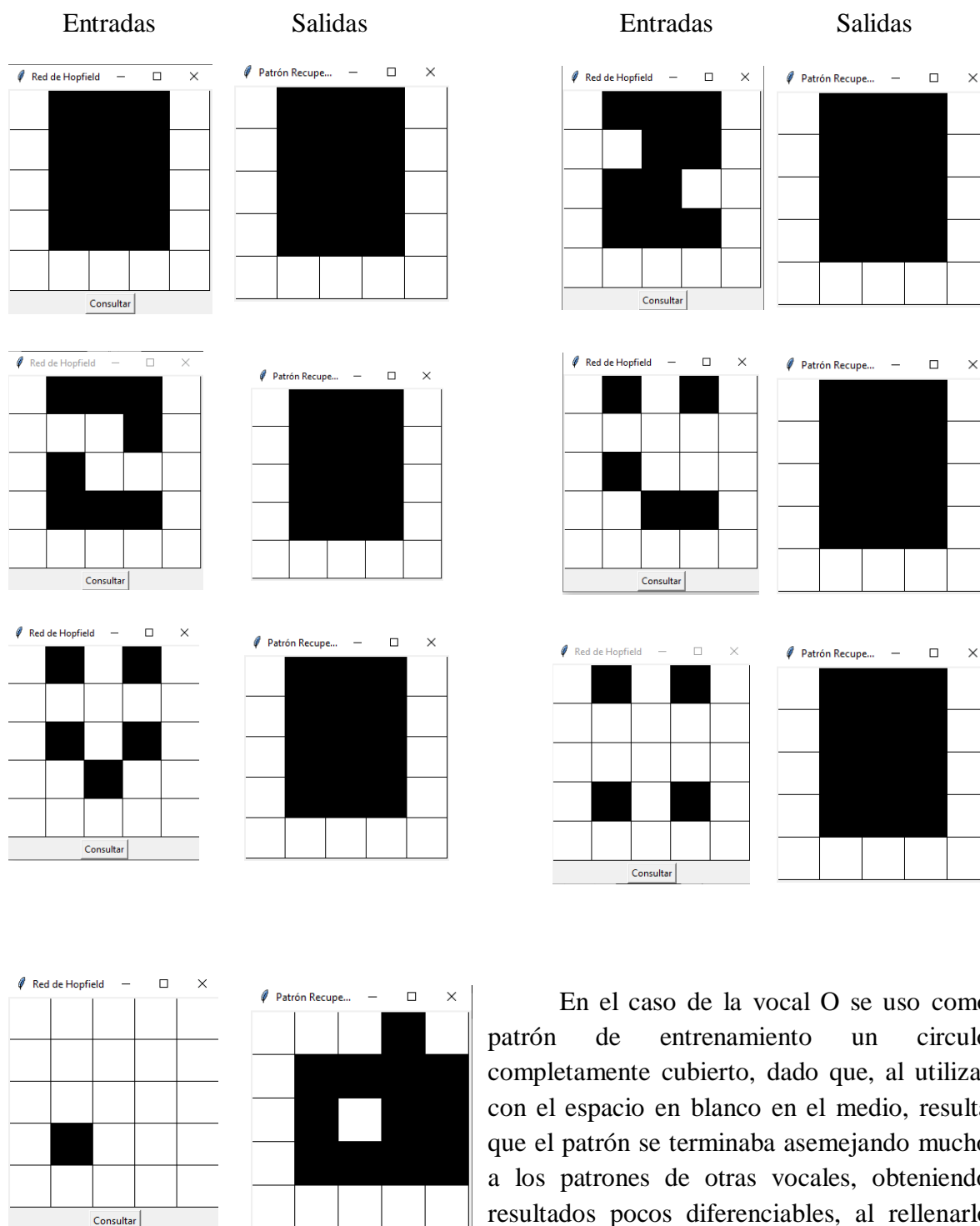


Salidas



Se observo, que siempre que se suministre información relevante, la red es capaz de entender a que patrón se hace referencia, existiendo puntos que tienen mucha mayor relevancia, en cuanto a su asociación de esos punto con respecto a un patrón, y que para ciertos patrones, se puede llegar a generar ruido.

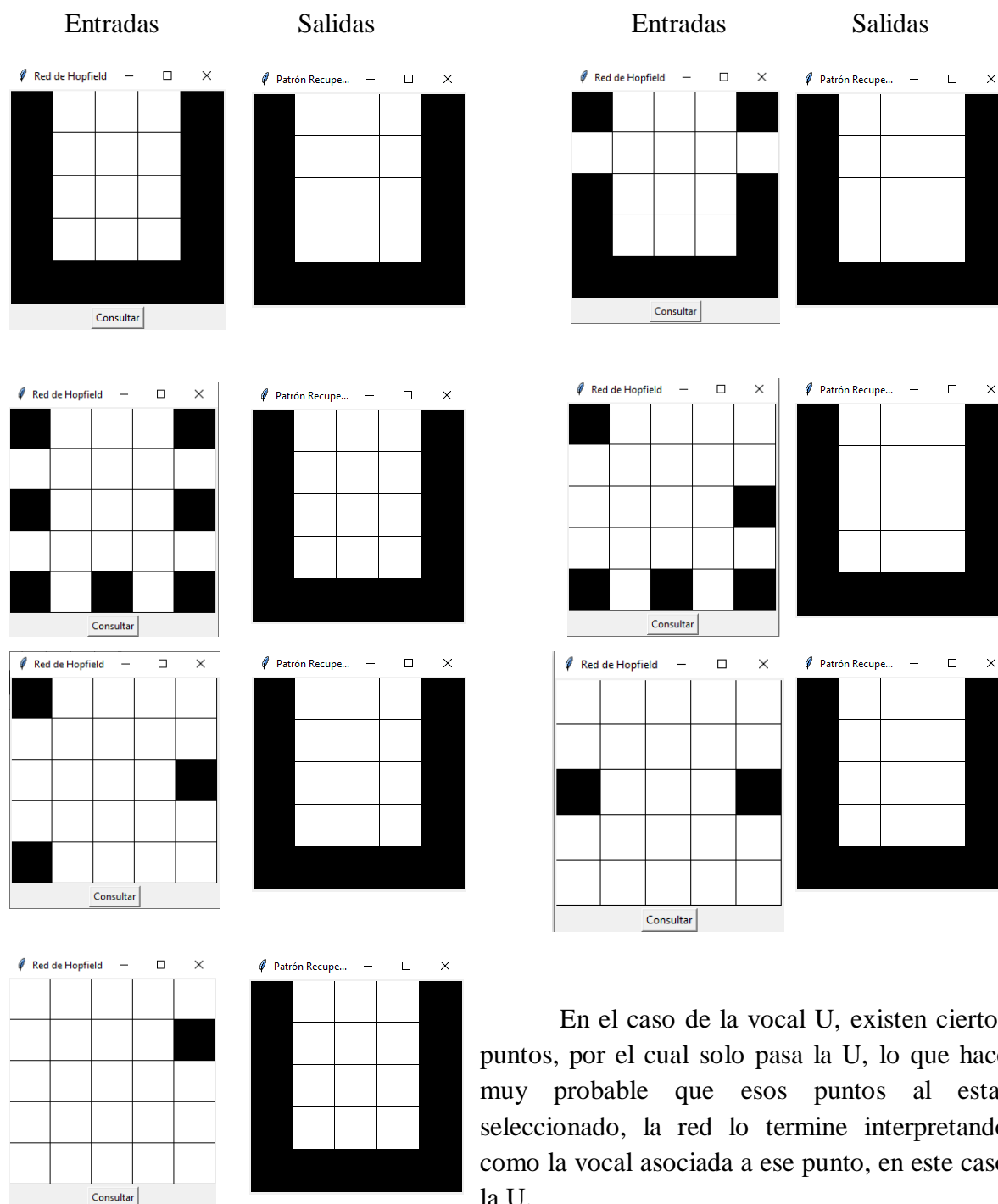
Pruebas con la vocal O:



En el caso de la vocal O se uso como patrón de entrenamiento un circulo completamente cubierto, dado que, al utilizar con el espacio en blanco en el medio, resulta que el patrón se terminaba asemejando mucho a los patrones de otras vocales, obteniendo resultados pocos diferenciables, al rellenarlo

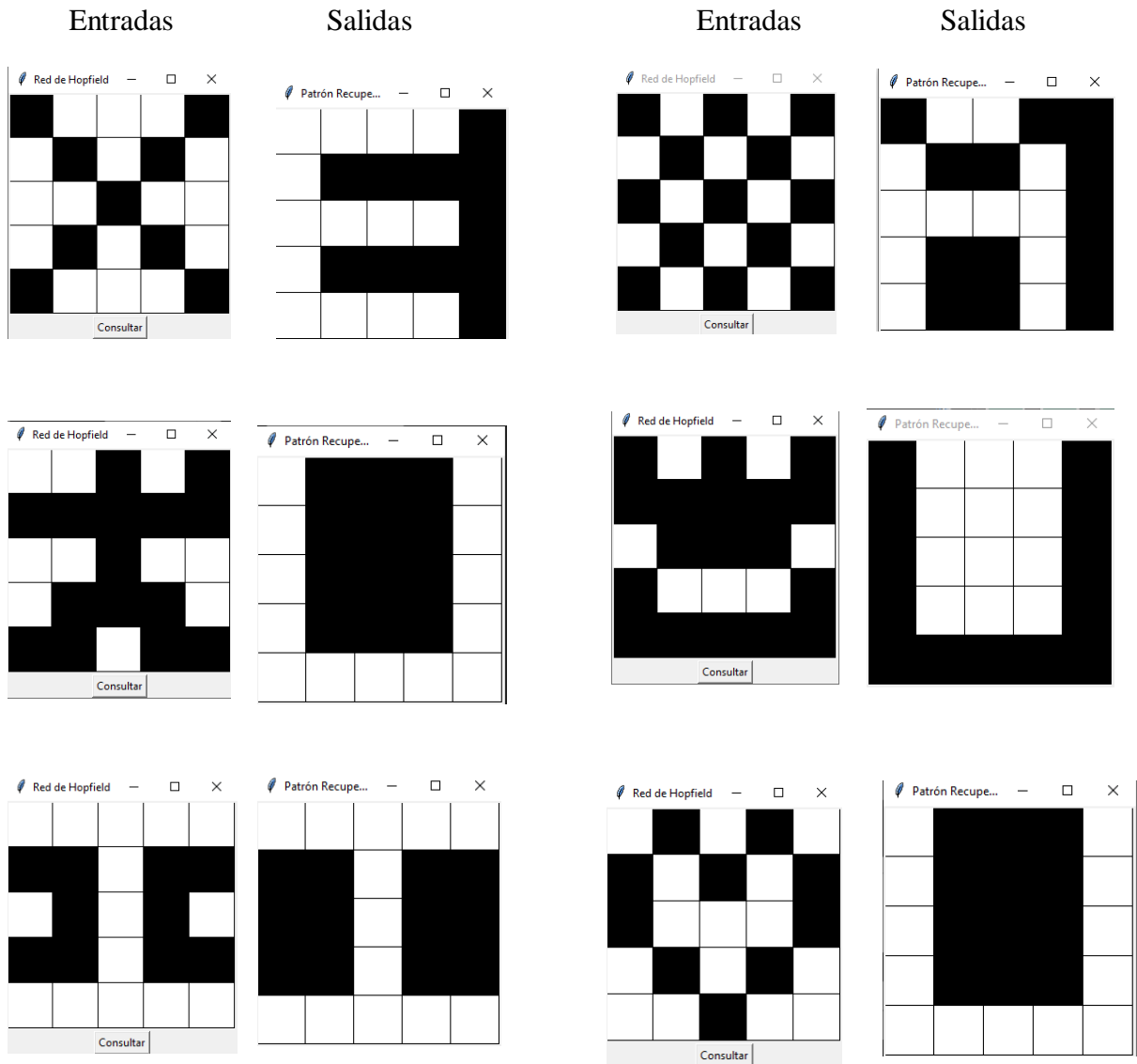
por completo hace que el patrón con el que se entreno tenga mayores diferencias, haciendo que se puedan diferenciar mejor los resultados.

Pruebas con la vocal U:



En el caso de la vocal U, existen ciertos puntos, por el cual solo pasa la U, lo que hace muy probable que esos puntos al estar seleccionado, la red lo termine interpretando como la vocal asociada a ese punto, en este caso la U.

Pruebas con patrones aleatorios:



Se puede observar como en la mayoría los casos los patrones resultantes de las consultas, se asemejan a los patrones con los que se realizó el entrenamiento, aunque en algunos casos se obtuvo su versión en negativo, igualmente se logra apreciar el patrón, pero es curioso ver como también se puede consultar por el inverso del patrón y obtener un resultado.



## Conclusiones

La red de Hopfield es un tipo de red neuronal recurrente que se utiliza para el almacenamiento y la recuperación de patrones, puede recuperar patrones completos a partir de estímulos parciales o ruidosos, aunque si la información suministrada contiene mucho ruido el patrón recuperado puede no ser el adecuado, esto puede variar con la utilización de una matriz de mayor tamaño para una mejor precisión, pero también requiere de un mayor poder computacional para realizar su entrenamiento.

El entrenamiento de la red de Hopfield implica actualizar los pesos de las conexiones sinápticas en función de los patrones de entrenamiento, esto se logra mediante una suma ponderada de las correlaciones de activación entre las neuronas.

La normalización de los pesos ayuda a equilibrar la influencia de los patrones de entrenamiento y garantiza que los pesos estén en un rango adecuado. En este proyecto, se normalizan los pesos dividiéndolos por el número total de patrones de entrenamiento.

La interfaz gráfica proporciona una forma interactiva de ingresar patrones de prueba y consultar la red de Hopfield para recuperar los patrones correspondientes, el usuario puede marcar y desmarcar las celdas en una cuadrícula y luego realizar la consulta.

La recuperación de patrones en la red de Hopfield se basa en la actualización iterativa de las activaciones neuronales hasta que el patrón de salida se estabilice, la convergencia se verifica mediante la comparación del patrón actual con el patrón anterior.

Este proyecto ofrece una implementación básica de la red de Hopfield y una interfaz gráfica funcional. Sin embargo, vale la pena resaltar, que se lograron apreciar comportamientos inesperados dentro de la red, al llegar a generar patrones que no formaban parte del set de entrenamiento, es interesante ver como en una única capa, se puede almacenar múltiples patrones dentro de la misma red, aunque conforme aumente el número de patrones que se desea recordar, la precisión puede variar. En este proyecto se lograron almacenar 5 patrones distintos dentro de la red

## **Bibliografía**

Maldonado Luis(2012).*Identificación Automática de Cilindros de Almacenamiento de Gas*

*Utilizando Redes Neuronales Tipo Hopfield.* Recuperado de:

<https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/103-111/4568>