Red Hopfield - Reconocimiento de Imágenes

Mauricio Tumalan Castillo - A01369288

***Resumen*— Este documento presenta el trabajo realizado para obtener una simulación de una red neuronal de Hopfield, la cual es utilizada para el reconocimiento de patrones binarios representados en imágenes. Se implementa un programa de Python que entrena la red utilizando un conjunto de imágenes seleccionadas y prueba su capacidad para recuperar patrones originales a partir de entradas ruidosas. Los resultados muestran que la red puede recuperar exitosamente los patrones aprendidos incluso cuando los datos de entrada contienen ruido.**

***Palabras Clave*— Red neuronal de Hopfield, Reconocimiento de patrones, Aprendizaje Hebbiano, Recuperación de patrones ruidosos.**

I. Introducción

Las redes neuronales de Hopfield son un tipo de red recurrente utilizada principalmente para almacenar patrones y recuperarlos de manera estable a partir de entradas distorsionadas. En este trabajo, se implementa una red de Hopfield para entrenar un conjunto de patrones de imágenes y evaluar su capacidad para converger hacia los patrones correctos incluso cuando las entradas contienen ruido. Esta simulación tiene como objetivo demostrar los principios de convergencia y estabilidad en redes neuronales de Hopfield.

II. Metodología

A. Selección del Conjunto de Símbolos

Se seleccionaron ocho imágenes del conjunto de datos disponibles en la librería skimage, las cuales representan patrones binarios complejos y fáciles de distinguir visualmente. Las imágenes incluyen figuras como un astronauta, un tablero de ajedrez, una moneda, la luna, entre otras. Cada imágen fue convertida a una representación en escala de grises, y posteriormente binarizada para cumplir con los requisitos de la red de Hopfield.

Figura 1: Imágenes originales

B. Preprocesamiento de las imágenes

El preprocesamiento de las imágenes consistió en los siguientes pasos:

1. Conversión a Escala de Grises: Las imágenes seleccionadas que estaban en formato RGB fueron convertidas a escala de grises utilizando la función rgb2gray de skimage, para reducir la dimensionalidad y simplificar el proceso de aprendizaje de la red.
2. Redimensionamiento: Cada imágen fue redimensionada a 128x128 píxeles mediante la función resize de skimage, generando un total de 16,384 píxeles por imagen, los cuales posteriormente se aplanaron para formar vectores unidimensionales.
3. Umbralización: Paso crucial en el preprocesamiento, se realizó utilizando la función threshold_mean. Esta función calcula un valor de umbral medio a partir de los valores de intensidad de los píxeles en la imagen. Los píxeles por encima del umbral se asignaron a 1, y los píxeles por debajo del umbral se asignaron a -1. Este proceso convierte las imágenes a un formato binario, necesario para el funcionamiento de la red de Hopfield.

   $$shift = 2 \times (binary * 1) - 1$$

   Esta fórmula asegura que los valores resultantes de los píxeles se ajusten al rango esperado de {-1.1}. El valor 1 representa un píxel activo y el -1 a un píxel inactivo, lo que convierte las imágenes en patrones binarios listos para ser procesados por la red Hopfield.
4. Aplanamiento de los Datos: Una vez binarizadas, las imágenes fueron aplanadas en vectores unidimensionales para poder ser utilizadas en la red Hopfield. Esto permitió que las imágenes pudieran ser manipuladas como vectores de estados neuronales en lugar de matrices de píxeles.

C. Entrenamiento de la Red Hopfield

La red Hopfield fue entrenada utilizando la regla de aprendizaje Hebbiano, adaptando los pesos neuronales de acuerdo con los patrones binarios generados. El proceso de entrenamiento incluyó los siguientes pasos:

1. Inicialización de la Matriz de Pesos: Se inicializa una matriz de pesos de dimensiones $N \times N$, donde N es el número de neuronas, equivalente al número de píxeles de cada imágen. Todos los pesos fueron inicialmente establecidos a cero.
2. Aprendizaje de Hebb: Para cada patrón de entrenamiento, se calculó la diferencia entre el patrón binarizado y un valor promedio ρ, que representa la media de todos los valores de los patrones. A continuación, se actualizó la matriz de pesos utilizando el producto exterior entre el patrón ajustado y sí mismo. Este método asegura que los pesos reflejan la relación entre las neuronas.
3. Normalización de los Pesos: Finalmente, la matriz de pesos fue normalizada dividiéndola por el número de patrones de entrenamiento, para asegurar una contribución balanceada de cada patrón.

D. Pruebas con Patrones Ruidosos

Con el objetivo de evaluar la capacidad de recuperación de patrones en la red Hopfield, se introdujo un nivel de ruido en los patrones de prueba. El ruido fue generado de la siguiente manera:

1. Generación de Ruido: A cada patrón de prueba se le aplicó un ruido del 20% y 40%, lo que significa que el 20% y 40% de los píxeles en cada imagen fue invertido aleatoriamente, cambiando de 1 a -1 o viceversa.
2. Evaluación de la Red: La red Hopfield fue utilizada para predecir los patrones originales a partir de las versiones ruidosas. Esto se logró mediante un proceso iterativo donde la red ajustaba los estados neuronales utilizando la regla de activación $sign(W \cdot x - \theta)$, donde W es la matriz de pesos, x el patrón de entrada y θ el umbral de activación.
3. Pruebas Asíncronas: Además, se evaluó el rendimiento de la red en modo asíncrono, en el cual las neuronas se actualizan de manera secuencial y aleatoria, en lugar de de forma simultánea.

E. Visualización de Resultados

Se generaron gráficos comparativos para visualizar los resultados de la red Hopfield. Para cada patrón entrenado, se mostró:

- El patrón original utilizado para el entrenamiento
- El patrón ruidoso que se utilizó como entrada a la red
- El patrón recuperado por la red tras el proceso de convergencia

III. Resultados

Los resultados muestran que la red neuronal de Hopfield es capaz de recuperar los patrones originales con alta precisión, incluso cuando el 20% de los datos de entrada estaban corrompidos así como con el 40%. En la Figura 2 y en la Figura 3 se presentan ejemplos de patrones originales, datos de entrada con ruido y patrones recuperados. En todos los casos, la red converge hacia el patrón entrenado, demostrando su capacidad de almacenamiento estable y recuperación.

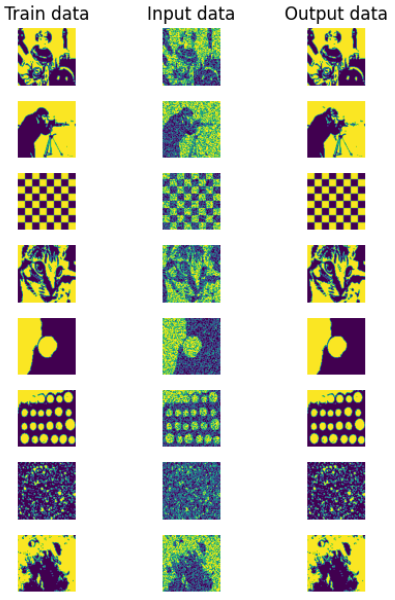

Figura 2: Resultados de la red con un 20% de ruido

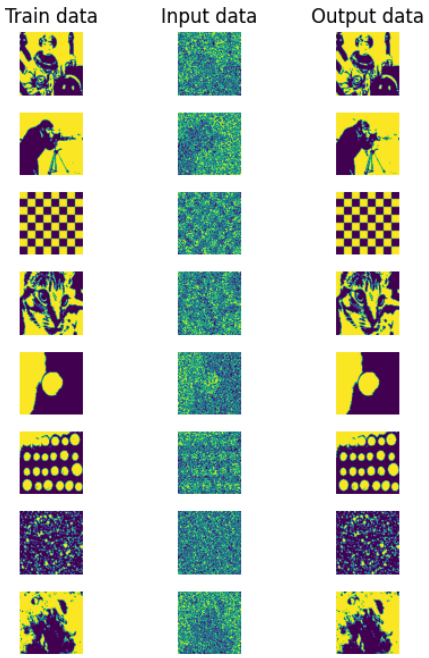

Figura 3: Resultados de la red con un 40% de ruido

Se puede observar que los patrones son recuperados satisfactoriamente, al agregar el patrón de un caballo, se puede observar en la Figura 4 que el modelo comienza a arrojar patrones no reconocibles para los patrones 2, 5 y 9.

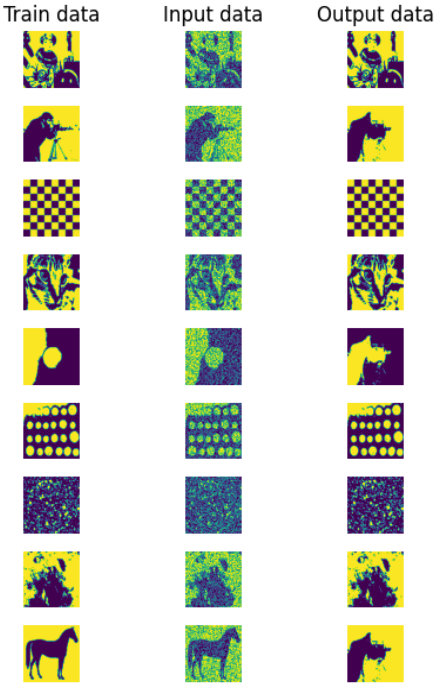

Figura 4: 9 patrones con 20% de ruido, mostrando errores

En la siguiente tabla, se pueden observar los resultados de la red:

Figura	Cambios	Parámetros Usados	Resultados	Comentario
1	Modelo base	Número de patrones: 8 Tasa de ruido: 20% Iteraciones del modelo permitidas para identificar el patrón: 20	Recuperación de los 8 patrones exitosa	Buen desempeño inicial
2	Modificación en la tasa de ruido	Número de patrones: 8 Tasa de ruido: 40% Iteraciones del modelo permitidas para identificar el patrón: 20	Recuperación de los 8 patrones exitosa	Buen desempeño a pesar del incremento de ruido
3	Modificación en el número de patrones	Número de patrones: 9 Tasa de ruido: 20% Iteraciones del modelo permitidas para identificar el patrón: 20	Recuperación de 6 de 9 patrones exitosa	Mientras más datos existan, más se le dificulta a la red recuperar los patrones

Tabla 1: Agrupación de resultados

IV. Conclusiones

La red neuronal de Hopfield mostró ser eficaz en la recuperación de patrones binarios almacenados, incluso en condiciones de ruido significativo. La simulación confirmó los principios de convergencia y estabilidad inherentes a este tipo de redes. Futuras investigaciones podrían enfocarse en ampliar el número de patrones almacenados y evaluar los límites de la capacidad de almacenamiento de la red.

Referencias

[1] T. Yamamoto, "Hopfield Network," GitHub Repository, [Online]. Available: https://github.com/takyamamoto/Hopfield-Network. [Accessed: Sep. 6, 2024].

[2] "Hopfield network," Wikipedia, Aug. 30, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Hopfield_network. [Accessed: Sep. 6, 2024].

[3] "Hopfield Neural Network," GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/hopfield-neural-network/. [Accessed: Sep.