*PROYECTO DE FORMATIVO*

*Sistema de Reconocimiento de Imágenes con Red de Hopfield*

UPDS SEDE LA PAZ

TURNO MAÑANA

Docente: Ing. Neysa Sandra Santalla Davalos

Estudiante: Israel Gonzalo Mariaca Torrez

Julio, 2024

La Paz, Bolivia

# 1. Introducción

## 1.1 Objetivo del proyecto

El objetivo principal de este proyecto fue desarrollar un sistema de reconocimiento de imágenes utilizando redes de Hopfield. Buscábamos crear una herramienta que no solo fuera funcional, sino también interactiva y fácil de usar para personas sin conocimientos técnicos profundos en redes neuronales.

## 1.2 Descripción general del sistema

Nuestro sistema permite a los usuarios entrenar una red de Hopfield con múltiples imágenes y luego utilizar esta red para reconocer patrones en nuevas imágenes. La interfaz gráfica proporciona controles para ajustar el preprocesamiento de las imágenes y visualizar tanto el proceso como los resultados del reconocimiento.

## 2. Fundamentos Teóricos

## 2.1 Redes de Hopfield

Las redes de Hopfield son un tipo de red neuronal recurrente capaz de almacenar y recuperar patrones. Funcionan como una memoria asociativa, donde los patrones se almacenan en las conexiones entre las neuronas. Cuando se presenta un patrón incompleto o ruidoso, la red puede converger a uno de los patrones almacenados, realizando así una tarea de reconocimiento o corrección de patrones.

## 2.2 Reconocimiento de patrones

En nuestro contexto, el reconocimiento de patrones implica identificar similitudes entre una imagen de entrada y los patrones previamente aprendidos por la red. Este proceso es fundamental en aplicaciones como el reconocimiento facial, la restauración de imágenes dañadas o la clasificación de objetos.

## 3. Implementación

## 3.1 Lenguaje y bibliotecas utilizadas

Elegimos Python como lenguaje de programación debido a su versatilidad y la amplia disponibilidad de bibliotecas científicas. Las principales bibliotecas utilizadas fueron:

- NumPy: para operaciones matriciales eficientes.

- Pillow (PIL): para el procesamiento de imágenes.

- Tkinter: para la creación de la interfaz gráfica.

- Matplotlib: para la visualización de gráficos.

## 3.2 Estructura del código

El código se organizó en varias clases y funciones:

- Clase HopfieldNetwork: implementa la lógica de la red de Hopfield.

- Funciones de utilidad: para cargar y guardar imágenes.

- Clase Application: maneja la interfaz gráfica y la interacción con el usuario.

## 3.3 Funcionalidades principales

Las funcionalidades clave incluyen:

- Entrenamiento de la red con múltiples imágenes.

- Reconocimiento de patrones en nuevas imágenes.

- Ajuste de parámetros de preprocesamiento de imágenes.

- Visualización de imágenes de entrada y reconocidas.

- Gráfico de evolución de la energía durante el reconocimiento.

## 4. Interfaz de Usuario

## 4.1 Diseño de la interfaz gráfica

La interfaz se diseñó pensando en la usabilidad, con una disposición clara de los controles y áreas de visualización. Se dividió en tres secciones principales: controles, visualización de imágenes y gráfico de energía.

## 4.2 Componentes principales

- Panel de control: botones para entrenar, reconocer, guardar y cargar la red.

- Sliders para ajustar contraste, brillo y nitidez.

- Áreas de visualización para imágenes de entrada y reconocidas.

- Gráfico para mostrar la evolución de la energía.

- Barra de progreso para operaciones largas.

## 5. Funcionalidades Implementadas

## 5.1 Entrenamiento de la red

Permite al usuario seleccionar múltiples imágenes para entrenar la red. Cada imagen se preprocesa según los parámetros actuales antes de ser utilizada para el entrenamiento.

## 5.2 Reconocimiento de imágenes

El usuario puede seleccionar una imagen para ser reconocida por la red entrenada. El sistema muestra tanto la imagen de entrada como el resultado del reconocimiento.

## 5.3 Preprocesamiento de imágenes

Implementamos ajustes de contraste, brillo y nitidez para mejorar la calidad de las imágenes antes del entrenamiento y reconocimiento.

## 5.4 Visualización de resultados

Además de mostrar las imágenes, el sistema genera un gráfico que muestra cómo cambia la energía de la red durante el proceso de reconocimiento, lo que ayuda a entender la convergencia del sistema.

## 5.5 Persistencia de datos

Se añadió la capacidad de guardar y cargar redes entrenadas, permitiendo a los usuarios conservar su trabajo entre sesiones.

6. Resultados y Discusión

## 6.1 Rendimiento del sistema

El sistema mostró un buen rendimiento en el reconocimiento de patrones simples y bien definidos. Sin embargo, su eficacia disminuye con patrones más complejos o cuando hay mucho ruido en las imágenes.

6.2 Limitaciones y posibles mejoras

- La red de Hopfield tiene limitaciones en cuanto a la cantidad y complejidad de los patrones que puede almacenar eficazmente.

- El sistema actual trabaja solo con imágenes en blanco y negro y de tamaño fijo (32x32 píxeles).

- Posibles mejoras incluyen la implementación de técnicas de reducción de dimensionalidad o la exploración de otros tipos de redes neuronales para el reconocimiento de imágenes más complejas.

7. Conclusiones

Este proyecto demostró la viabilidad de utilizar redes de Hopfield para el reconocimiento de patrones en imágenes, proporcionando una interfaz accesible para usuarios no técnicos. Aunque el sistema tiene limitaciones, sirve como una excelente herramienta educativa para entender los principios básicos del reconocimiento de patrones y las redes neuronales recurrentes.

8. Referencias

- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79(8), 2554-2558.

- Rojas, R. (1996). Neural networks: a systematic introduction. Springer Science & Business Media.

- Python Software Foundation. Python Language Reference, version 3.x. Available at http://www.python.org