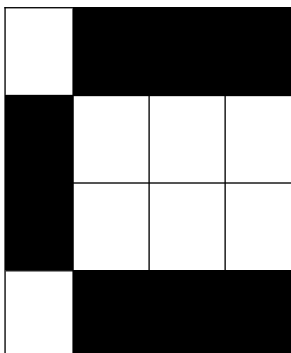
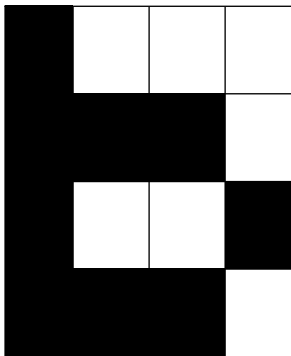
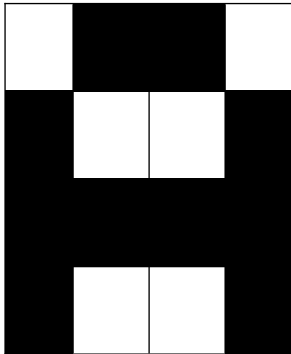


TP 2 2025

Desarrollar una red neuronal de Hopfield

Se debe construir desde cero una red neuronal artificial de Hopfield del tipo discreta. Esta red debe poder reconocer imágenes en blanco y negro que representan las letras del abecedario, por ejemplo:



Se espera que por lo menos sea capaz de reconocer 10 letras del total de las 27 que representan al abecedario en español.

Para representar cada letra (de las que se vaya a reconocer) se debe construir una cuadrícula de 4x4 elementos donde cada celda de dicha cuadrícula puede ser de color blanco o negro.

La red neuronal a construir debe poder manejar dos conjuntos de vectores de entrada: un conjunto será utilizado para entrenamiento y otro para test y reconocimiento. Es decir que la red debe ser capaz de entrenar y reconocer imágenes finalmente. Se aclara que en el conjunto de test debe haber imágenes parecidas pero no iguales a las de entrenamiento.

El programa a desarrollar debe ser capaz de entrenar por medio de un conjunto de imágenes del tipo antes mencionado, generando y mostrando por pantalla la matriz de pesos W . **Se aclara que el conjunto de imágenes de entrenamiento debe ser reemplazado por los vectores del tipo:**

-1	1	1	-1	1	-1	-1	1	1	1	1	1	1	-1	-1	1
----	---	---	----	---	----	----	---	---	---	---	---	---	----	----	---

(Este vector, por ejemplo, representa a la letra "A" donde 1 representa el valor negro y el -1 un valor blanco)

Además dicho programa debe contar con la capacidad para poder reconocer las imágenes de entrenamiento y test indistintamente. Por otro lado el programa debe ser capaz de saber si logra o no converger al reconocimiento de dicha imagen mostrando los siguientes resultados por pantalla:

- El producto correspondiente de cada entrada de test con la matriz W en cada paso.
- Valores de la función activación en cada paso.

Se debe tener en cuenta que al realizar una predicción la red debe mostrar la salida del vector correspondiente y también la imagen que representa dicho vector.

Se espera:

1. Que la aplicación corra sin presentar errores y que se entregue un documento de respaldo con las explicaciones pertinentes sobre el trabajo realizado. **(20 ptos)**
2. Que la aplicación cuente con una interfaz amigable que permita: **(25 ptos)**
 - a. Poder consumir imágenes como entrada y no vectores (por supuesto luego la imagen debe ser transformada en vector para que la red la pueda procesar).
 - b. Ver las imágenes de entrenamiento y test.
 - c. Ver los vectores de entrenamiento y test.
 - d. Transformar una imagen en vector y viceversa.

- e. El producto correspondiente de cada entrada de test con la matriz W en cada paso.
 - f. Valores de la función activación en cada paso.
3. Que la aplicación permita agregar más las letras que las mínimas solicitadas en el enunciado o que la red aprenda a reconocer y entrenar con imágenes de mayor tamaño de cuadrícula. **(25 ptos)**
 4. Que la aplicación permita “reconocer” instancias totalmente desconocidas (que nunca fueron entrenadas). **(15 ptos)**
 5. Determinar (al menos desde un punto de vista teórico) que sucede cuando los patrones de entrenamiento son muy similares. Se espera que se propongan caminos (o explicaciones) de solución. **(15 ptos)**