

Red Neuronal de Hamming

1. Introducción

La red neuronal de Hamming es una red de tipo autoasociativa utilizada para el reconocimiento de patrones.

Funciona comparando un vector de entrada con un conjunto de patrones almacenados y seleccionando el patrón más similar.

2. Objetivo del Proyecto

Implementar una red neuronal de Hamming en Python, capaz de identificar el patrón más similar entre una serie de vectores binarios de entrada, aplicado a un contexto relacionado con estudiantes de Ingeniería de Sistemas.

3. Estructura del Proyecto

```
red-hamming/  
├── data/  
│   └── entradas.csv  
├── src/  
│   ├── main.py  
│   └── utils.py  
├── README.md  
└── patrones.csv
```

4. Requisitos del Sistema

- Python 3.10 o superior
- Pandas

5. Instalación y Ejecución

1. Crear entorno virtual (opcional pero recomendado):

```
python -m venv venv  
venv\Scripts\activate
```

2. Instalar dependencias:

```
pip install pandas
```

```
python src/main.py
```

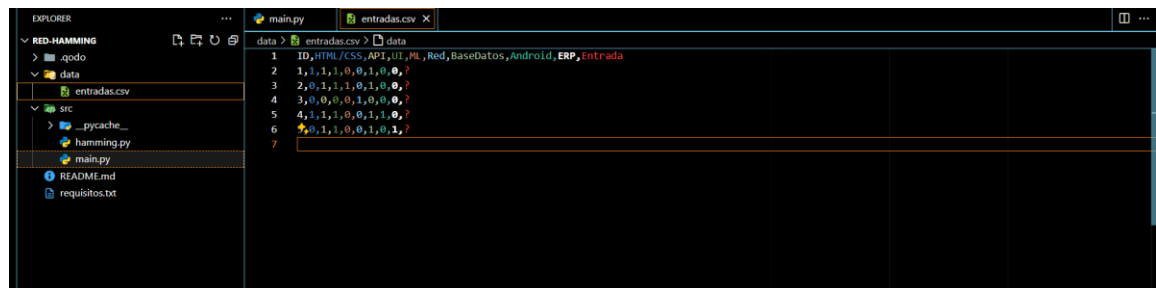
El programa toma un archivo CSV (`entradas.csv`) con vectores binarios. Luego compara cada vector con un conjunto de patrones de referencia.

7. Ejemplo de Aplicación en Ingeniería de Sistemas

La red puede ayudar a clasificar a los estudiantes según el tipo de perfil profesional más cercano (desarrollador frontend, backend, analista, etc.).

- `entradas.csv`: contiene los vectores de entrada a evaluar.
- `patrones.csv`: contiene los patrones de referencia (perfiles tipo).
- `main.py`: ejecuta la red de Hamming y muestra el resultado de clasificación.

DATA



IMPLEMENTACION

```
1 import numpy as np
2
3
4 def hamming_network(W, x, threshold=2.5):  Rename this parameter "w" to match the regular expression ^[_a-z][a-z0-9_]*$.
5     h = np.dot(W, x)
6     f = np.where(h > threshold, 1, 0)
7     return h, f
8
```

```
22
23 with open('./data/entradas.csv') as f:
24     reader = csv.reader(f)
25     next(reader) # saltar cabecera
26     for i, row in enumerate(reader):
27         x = np.array(list(map(int, row[1:9])))
28         h, f = hamming_network(W, x)
29         if np.sum(f) == 0:
30             resultado = "Ninguno supera el umbral"
31         else:
32             resultado = proyectos[np.argmax(h)]
33         print(f"Entrada {i+1}: h = {h.tolist()} + {resultado}")
34
```

RESULTADOS

```
Entrada 2: h = [3, 3, 4, 0, 3] + Inteligencia Artificial
Entrada 3: h = [0, 0, 0, 1, 0] + Ninguno supera el umbral
Entrada 4: h = [4, 5, 3, 0, 3] + Aplicación Móvil
Entrada 5: h = [3, 3, 3, 0, 4] + Sistemas Empresariales

D:\ciclo-7-2025-1\ANALISIS MULTIVARIADO\UNIDAD 3\red-neuronal\red-hamming\
```