# Guía de Redes de Hopfield: Fundamentos y Aplicación a Recuperación de Claves

David Garcia Cano

## Introducción

Las redes de Hopfield son un tipo de red neuronal recurrente que puede almacenar patrones binarios y recuperarlos a partir de versiones incompletas o ruidosas. Este documento actúa como una guía teórica para su uso en el proyecto `neurocipher`, orientado a la recuperación de claves binarias.

## 1. Modelo matemático de la red de Hopfield

Una red de Hopfield clásica está formada por $N$ neuronas binarias $x_i \in \{-1, +1\}$, completamente conectadas entre sí, con pesos simétricos $w_{ij} = w_{ji}$ y sin autoconexiones $w_{ii} = 0$.

El estado global de la red es un vector:

$$\mathbf{x} = (x_1, x_2, \ldots, x_N)$$

### 1.1. Regla de Hebb para el aprendizaje

Si se desea almacenar un conjunto de patrones $\{\xi^\mu\}$ con $\mu = 1, \ldots, P$, cada patrón $\xi^\mu \in \{-1, +1\}^N$, los pesos sinápticos se definen como:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \quad \text{con } w_{ii} = 0$$

Esta fórmula garantiza que los patrones almacenados sean puntos de equilibrio del sistema.

### 1.2. Dinámica de actualización

La red evoluciona asíncronamente (o sincrónicamente) según la siguiente regla de actualización para cada neurona $i$:

$$x_i(t+1) = \text{sign}\left( \sum_{j=1}^{N} w_{ij} x_j(t) \right)$$

$$\text{sign}(z) = \begin{cases} +1 & \text{si } z > 0 \\ -1 & \text{si } z < 0 \\ x_i(t) & \text{si } z = 0 \end{cases}$$

Este proceso se repite hasta alcanzar un estado estable (mínimo de energía).

### 1.3. Energía del sistema

La red minimiza una función de energía definida como:

$$E(\mathbf{x}) = -\frac{1}{2}\sum_{i \neq j} w_{ij} x_i x_j$$

El proceso de actualización garantiza que la energía del sistema decrece o se mantiene constante en cada iteración.

## 2. Capacidad de almacenamiento

La red puede almacenar correctamente hasta aproximadamente $0{,}138N$ patrones ortogonales sin errores de recuperación. Más allá de ese límite, pueden aparecer errores debidos a interferencias (efecto crosstalk).

## 3. Aplicación en recuperación de claves binarias

En este proyecto, una clave binaria (por ejemplo, de 128 bits) puede representarse como un patrón $\mathbf{x} \in \{-1, +1\}^{128}$. La red de Hopfield se entrena con esa clave y, tras añadir ruido al patrón (invertir un porcentaje de bits), se prueba si la red es capaz de recuperar la clave original.

- Patrón original: clave de 128 bits convertida a $\{-1, +1\}$
- Patrón ruidoso: mismo vector con 20 % de los bits invertidos
- Patrón recuperado: resultado tras evolución de la red

## 4. Mejoras progresivas por versión

### Versión 1.0

- Red clásica de Hopfield con regla de Hebb.
- Actualización asíncrona bit a bit.
- Sin tolerancia al ruido significativa.

### Versión 1.1

- Se introduce un sistema de testeo con diferentes niveles de ruido.
- Evaluación de precisión y coincidencia exacta.
- Resultados sirven como línea base comparativa.

### Versión 1.2

- Se aplica entrenamiento con patrones ligeramente ruidosos (annealing inicial).
- Se añaden repeticiones del entrenamiento (`repetitions`).
- Introducción de actualización por bloques de bits (`chunk_size`).

### Versión 1.3-A

- Combinación de todas las técnicas anteriores: entrenamiento ruidoso, múltiples repeticiones, actualización por bloques.
- Evaluación sistemática de hiperparámetros (`annealing_start`, `repetitions`, `chunk_size`).
- Mejora drástica en precisión y robustez ante ruido alto.

## 5. Resultados experimentales

Se compararon las versiones anteriores mediante simulaciones con claves de 128 bits, evaluando la **precisión media** frente al **nivel de ruido**.

- **V1.0 y V1.1:** el rendimiento cae bruscamente a partir del 20–30 % de ruido.

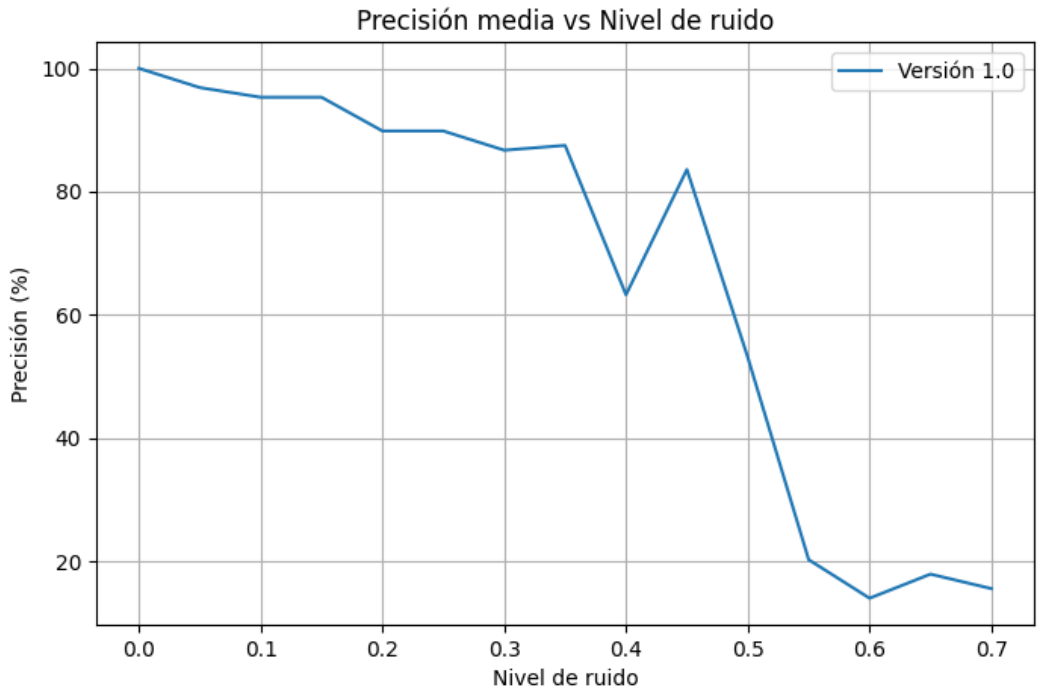


Figura 1: Recuperación de patrones en el estado incial de la red de Hopfield.

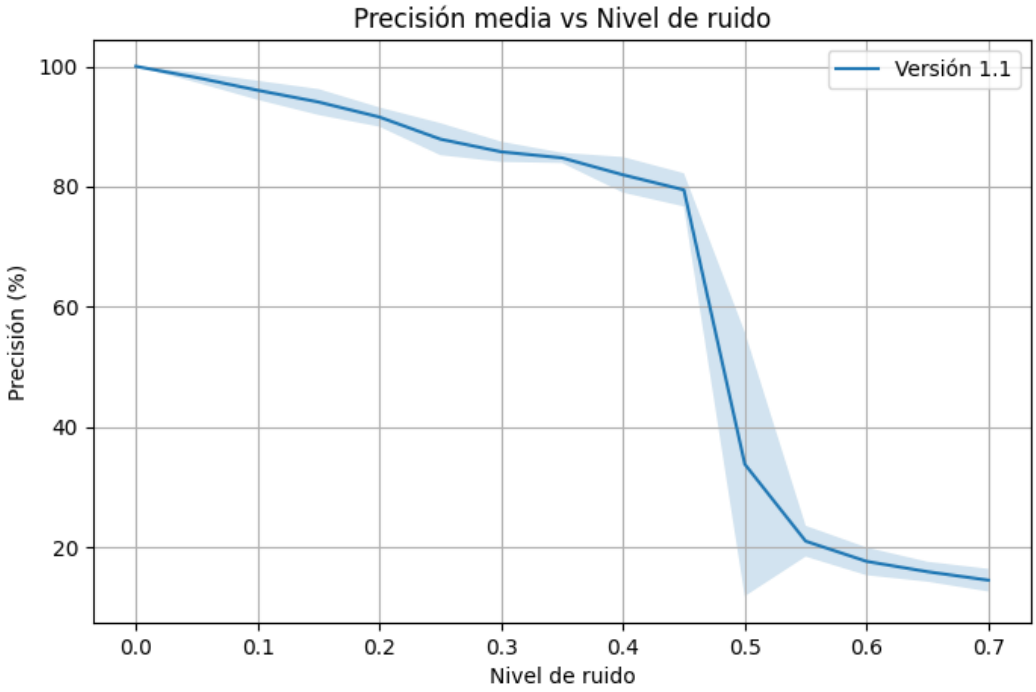


Figura 2: Recuperación tras la introdcción del annealing.

- **V1.2:** mejora significativa hasta el 40 %, pero limitada a configuraciones concretas.
- **V1.3-A:** mantiene alta precisión incluso por encima del 50 % de ruido.

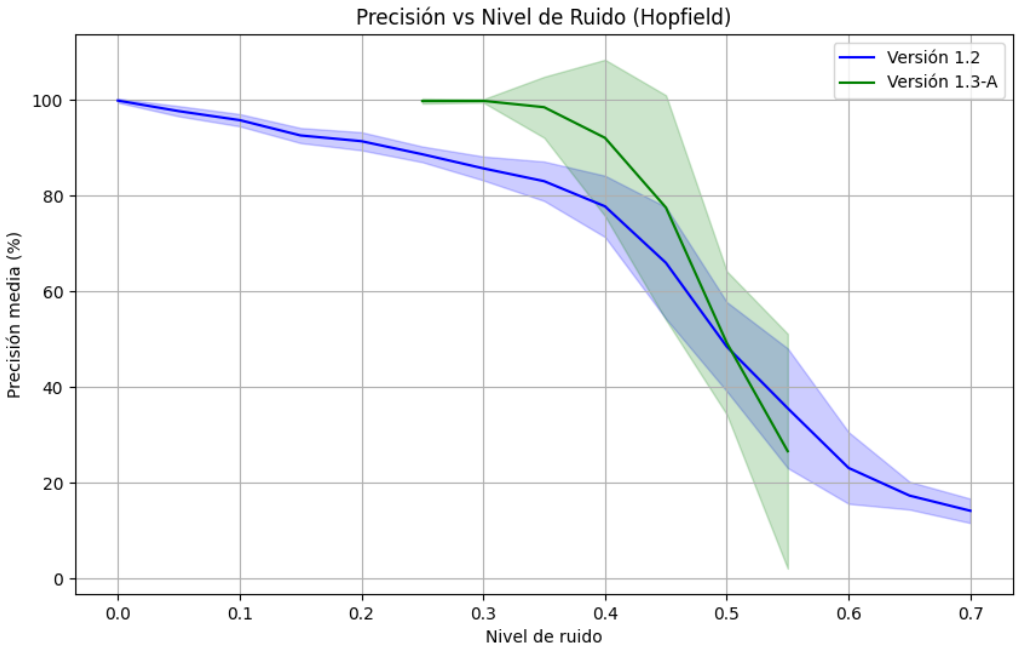


Figura 3: Comparativa de precisión media vs nivel de ruido. La versión 1.3-A supera a la 1.2 a partir del 30 % de ruido.

# 6. Cifrado y recuperación de claves

Además de probar la capacidad de recuperación de patrones binarios aislados, se implementó un sistema completo de cifrado y descifrado usando claves almacenadas en una red Hopfield. Este proceso forma parte de la FASE 5 del proyecto `neurocipher`, cuyo objetivo es integrar criptografía clásica con recuperación neuronal robusta.

## Metodología

- Se generó una clave cifrante (AES de 256 bits o RSA con $d$ de 128 bits).
- Se cifró un mensaje real usando dicha clave.
- La clave se convirtió a un patrón binario $\{-1, +1\}$ y se almacenó en una red Hopfield robustecida (annealing + repeticiones + chunked update).
- Se aplicó ruido al patrón (de 0 % a 60 % de inversión de bits).
- La red intentó recuperar el patrón original y, si era suficientemente preciso (ideal: 100 %), se usó para descifrar el mensaje.

## Resultados de cifrado con recuperación Hopfield

Los resultados muestran que la clave debe recuperarse **con precisión exacta** para que el descifrado tenga éxito. Por tanto, la tasa de éxito en descifrado es prácticamente equivalente a la tasa de precisión 100 % en la red.

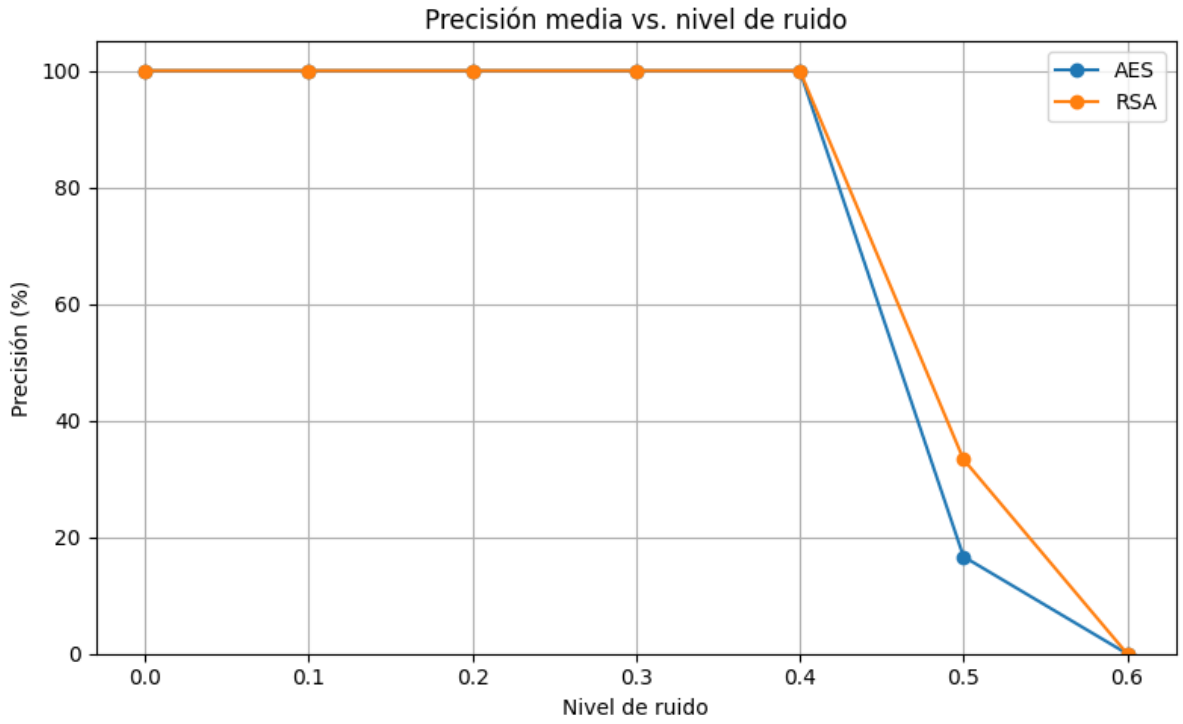


Figura 4: Precisión media en recuperación de claves cifrantes reales (AES y RSA) usando red Hopfield 1.3-A.

# 7. Fundamento de los cifrados utilizados

El sistema Hopfield no opera solo: se integra con sistemas criptográficos reales. La generación, cifrado y descifrado de claves se basa en tres pilares matemáticos documentados por separado:

- **RSA clásico:** generación de claves mediante primos grandes, inverso modular y funciones exponenciales. Implementado desde cero con funciones propias (`gcd`, `modinv`, etc.).
- **Cifrado simétrico con Fernet:** basado en AES-128-CBC + HMAC-SHA256, con autenticación y protección de integridad (esquema encrypt-then-MAC).
- **ECC y ECDSA:** curvas elípticas definidas sobre $\mathbb{F}_p$, operación de grupo y firma digital basada en multiplicación escalar segura.

Para una explicación detallada de cada esquema, se remite al documento técnico *Guía de funciones RSA y cifrados simétricos* incluido en el repositorio del proyecto.

# 8. Referencias básicas

- J.J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, *PNAS*, vol. 79, no. 8, pp. 2554–2558, 1982.

- Hertz, Krogh, Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.
- Haykin, S. *Neural Networks and Learning Machines*, Pearson, 2008.