

Redes-Neuronales - TP1



Nombre	Camila Fernández Marchitelli
Padrón	102515
Año	2025
Mail	cfernandezm@fi.uba.ar

Tabla de Contenidos

- [Introducción](#)
- [Red recurrente](#)
- [Modelo de Hopfield](#)
- [Trabajo Práctico - Parte 1](#)
 - [Punto a](#)
 - [Punto b](#)
 - [Punto c](#)
 - [Punto d](#)
- [Trabajo Práctico - Parte 2](#)
 - [Punto a](#)
 - [Punto b](#)
- [Trabajo Práctico - extra](#)

Introducción

¿Qué es una red neuronal biológica y como se imitan mediante algoritmos?

Red neuronal biológica: funciona con neuronas, impulsos eléctricos y sinapsis. Cada neurona decide si transmite una señal en función de la suma de sus entradas y la fuerza de las conexiones. El aprendizaje ocurre reforzando o debilitando esas conexiones.

Red neuronal artificial: está compuesta por “nodos” o “neuronas artificiales” que realizan operaciones matemáticas simples.

Las redes neuronales artificiales imitan de manera muy simplificada la forma en que las redes biológicas procesan y aprenden información. No replican la complejidad del cerebro, pero toman la idea de “muchas unidades simples conectadas en red que, al trabajar juntas, generan comportamientos complejos”.

Para que una red neuronal pueda funcionar correctamente necesita pasar por un proceso de entrenamiento, en el cual se ajustan los pesos de sus conexiones entre capas. Mediante la exposición a distintos conjuntos de datos, el sistema va perfeccionando su capacidad de aprendizaje, logrando que las neuronas artificiales reconozcan patrones y resuelvan problemas, ofreciendo respuestas que se asemejan a las de un ser humano.

Red recurrente

Una red recurrente es aquella en donde las neuronas tienen conexiones que vuelven sobre sí mismas o hacia otras neuronas en la misma capa, lo que permite que el sistema “recuerde” estados anteriores.

La red de Hopfield es una de ellas. En la próxima sección se hace una introducción al modelo

Modelo de Hopfield

El modelo de Hopfield, propuesto por John Hopfield en 1982, es una de las primeras arquitecturas de redes neuronales artificiales que buscó imitar el funcionamiento de la memoria en el cerebro

La idea principal es que la red pueda funcionar como una memoria asociativa:

- Durante el entrenamiento, se almacenan patrones en los pesos de la red.
- Cuando se le presenta un patrón incompleto o ruidoso, la red evoluciona dinámicamente hasta estabilizarse en el patrón correcto o en uno muy cercano.

Memoria asociativa

La memoria asociativa es la capacidad de un sistema de recordar información completa a partir de fragmentos o datos incompletos. En el cerebro biológico sería, por ejemplo: ver solo una parte de una foto y aun así reconocer a la persona que aparece.

En una red de Hopfield, esta idea se implementa almacenando patrones en los pesos de las conexiones.

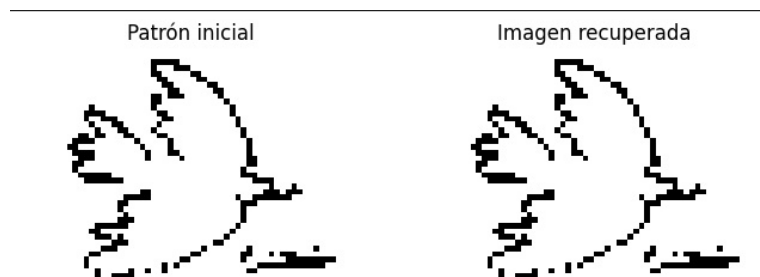
La memoria asociativa en Hopfield permite que la red actúe como un recordador de patrones, reconociendo información parcial y llevándola al patrón más parecido que ya tenía guardado.

Trabajo Práctico - Parte 1

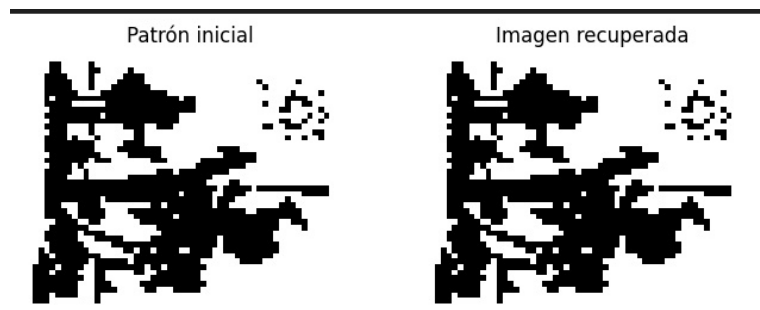
Entrene una red de Hopfield '82 con las imágenes binarias disponibles en el campus.

a- Verifique si la red aprendió las imágenes enseñadas.

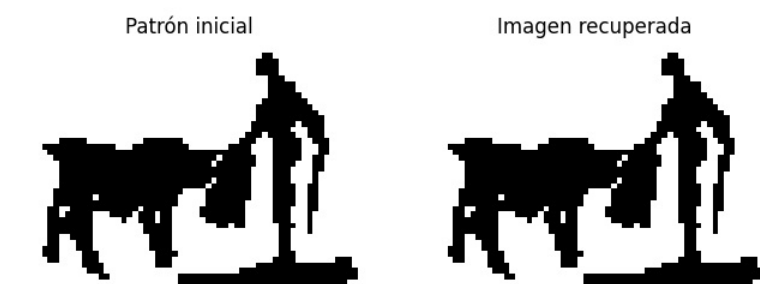
Para probar esto, le presento a la red 3 imágenes del mismo tamaño y evalúo la convergencia de cada imagen.



Patrón 1



Patrón 2



Patrón 3

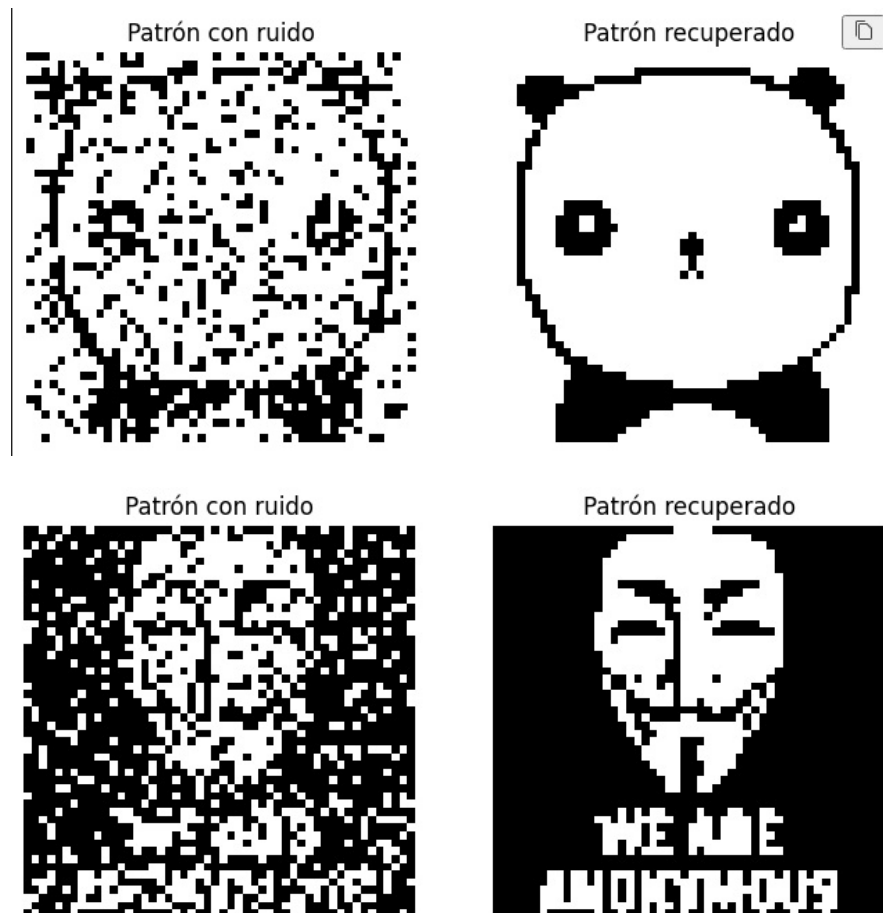
La imagen recuperada es la misma que el patrón inicial. Por lo que podemos concluir que la red aprendió de los patrones enseñados.

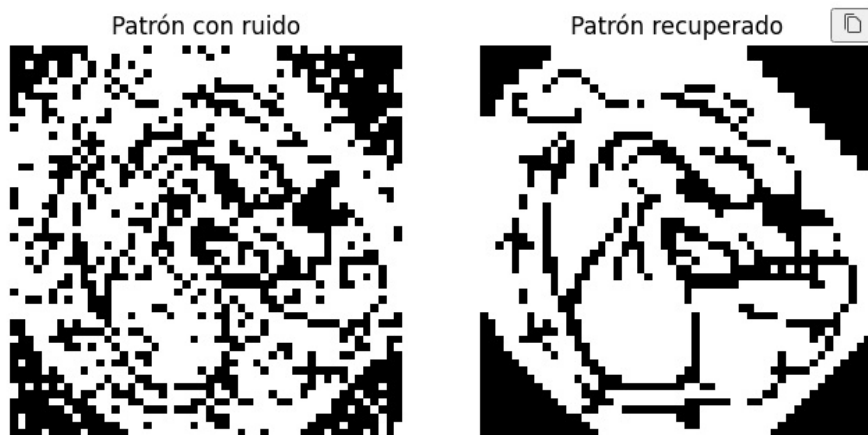
b- Evalúe la evolución de la red al presentarle versiones alteradas de las imágenes aprendidas: agregado de ruido, elementos borrados o agregados.

Agregandole ruido a los patrones invirtiendo los bits

Para este punto elegí las imágenes de tamaño (50,50), les agregué diferentes grados de ruido y probé la red. Elegí agregarle ruido invirtiendo bits de forma aleatoria.

Se puede ver que la red es bastante tolerante al ruido elegido, ya que logró converger a la imagen inicial en los 3 casos.



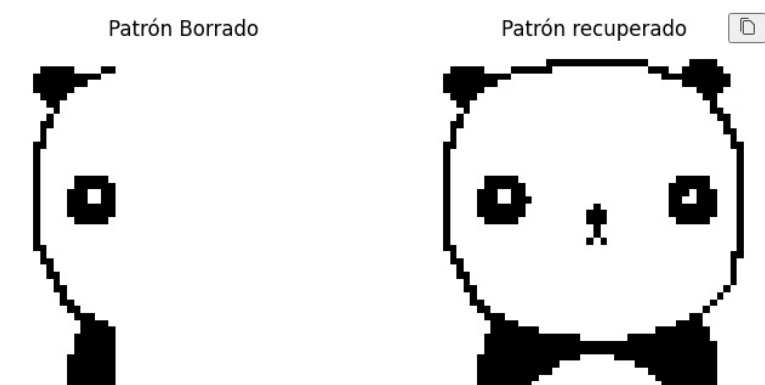


Agregándole ruido a los patrones borrando diferentes porcentajes del patrón

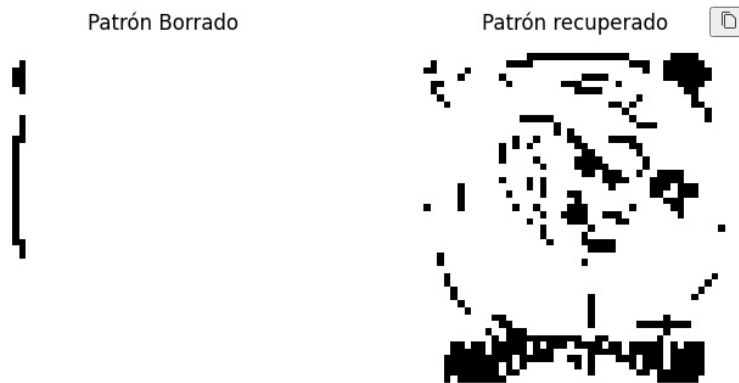
Se puede observar que borrando hasta un 70% del patrón original la red converge de forma correcta.



Borrando el 50% del patrón original



Borrando el 70% del patrón original



Borrando el 90% del patrón original

c- Evalúe la existencia de estados espurios en la red: patrones inversos y combinaciones de un número impar de patrones.

Estados espurios

Una red de Hopfield es una red de memoria asociativa: entrena ciertos patrones ξ_μ como mínimos de energía. Cuando a la red se le da una entrada ruidosa o incompleta, la dinámica baja la “energía” y cae en el mínimo más cercano es decir, recupera el patrón original. Además de estos patrones entrenados, aparecen otros mínimos estables que no son patrones originales, a esos mínimos se los llama estados espurios.

Tipos de estados espurios

1- Patrones inversos

Si ξ_μ es un patrón entrenado, su inverso $-\xi_\mu$ también puede ser un mínimo estable. Por ejemplo: Si la red memorizó un panda, la red puede converger también en un “anti panda” todo invertido lo blanco/negro.

2- Combinaciones de un número impar de patrones

Un estado espurio también puede ser una suma de k (número impar, observar en la formula $2k+1$) estados enseñados:

$$s = \text{sign} (\xi^{\mu_1} + \xi^{\mu_2} + \dots + \xi^{\mu_{2k+1}})$$

Es decir, si se entrenó la red con tres imágenes distintas, la red puede generar una especie de mezcla híbrida de esas tres y tratar de recordarla como si fuera un patrón real.

Visualmente, esas combinaciones suelen parecer “fantasmas” de varios patrones mezclados.

Como podemos ver esto en la practica

- Si la red se inicializa con un patrón muy ruidoso, en vez de converger al patrón correcto, puede caer en uno de estos estados espurios.
- A medida que se sobrecarga la red, los espurios se vuelven más frecuentes y limitan la utilidad de Hopfield.

Teniendo en cuenta lo anteriormente explicado, voy a evaluar la existencia de estados espurios en la red ya entrenada.

Elijo los patrones de tamaño (50x50) ya que tienen mas detalle de color.

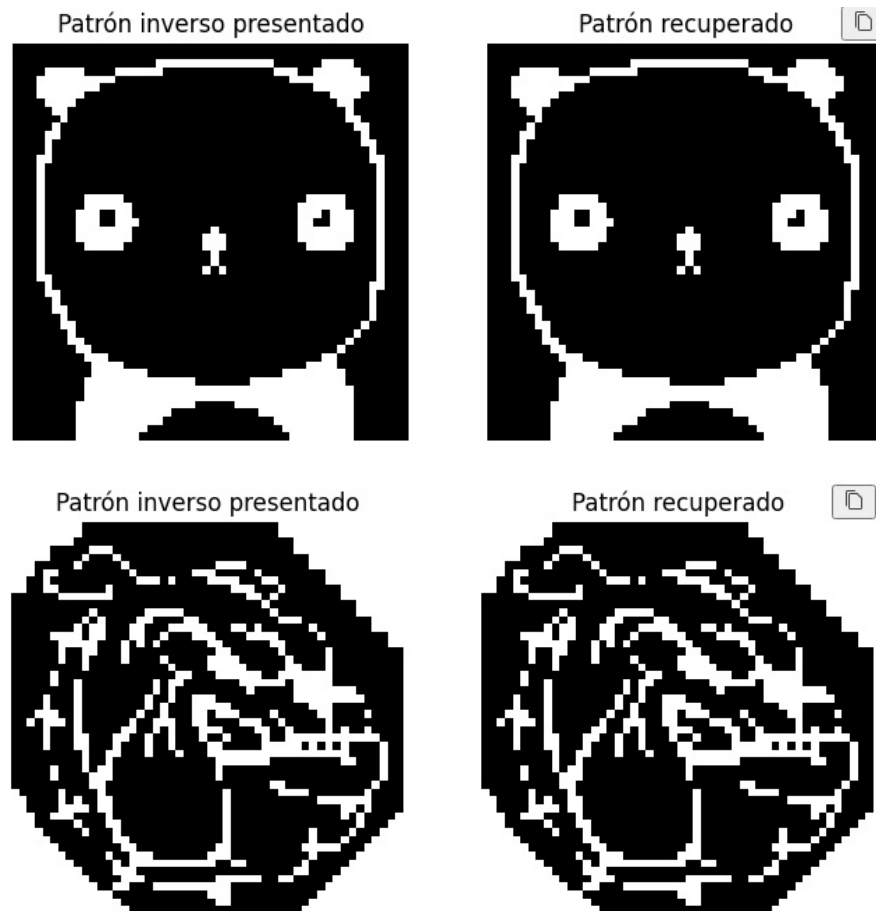
Lo que se hace es:

1. Entreno a la red con los patrones reales.
2. Le doy como entrada un patrón inverso (que nunca fue parte del entrenamiento).

Si la red:

1. Devuelve el patrón real entonces el inverso no es estable.
2. Si converge al patrón inverso entonces ese inverso se convirtió en un estado espurio.

Patrones inversos



Patrón inverso presentado



Patrón recuperado



Como pudimos observar estos patrones inversos son estados espurios de la red. La red recuerda también la versión invertida aunque no le haya enseñado antes porque matemáticamente tiene la misma coherencia interna que el patrón real.

Combinaciones de un número impar de patrones

Patrón combinado presentado



Patrón recuperado



Combinación con 3 patrones

Decidí tomar una combinación impar de patrones que no estaba en el set de entrenamiento y podemos observar que la red lo devolvió tal cual y no coincide con ninguno de los patrones enseñados. Por lo que se puede concluir que es un estado espurio del tipo “combinación impar de patrones”

d- Realice un entrenamiento con las 6 imágenes disponibles. ¿Es capaz la red de aprender todas las imágenes? Explique.

Para realizar el entrenamiento con todas las imágenes, lo primero que decido hacer es normalizarlas a todas en un mismo tamaño de 50x50 completando con píxeles blancos.

Patrón presentado



Patrón recuperado



Patrón presentado



Patrón recuperado



Patrón presentado



Patrón recuperado



Patrón presentado



Patrón recuperado



Patrón presentado



Patrón recuperado



Patrón presentado



Patrón recuperado



Se puede observar que luego de presentarse los 6 patrones con los que la red aprendió, no pudo aprender bien todas las imágenes, ya que en 4 casos no logró converger al patrón real.

Este comportamiento refleja las restricciones de las redes de Hopfield: su memoria es aproximada y no garantiza la recuperación perfecta de todos los patrones entrenados, especialmente si son visualmente parecidos o si se introducen en gran cantidad, ya que posiblemente se haya alcanzado su capacidad máxima de almacenamiento.

Trabajo Práctico - Parte 2

a- Comprobar estadísticamente la capacidad de la red de Hopfield '82 calculando la cantidad máxima de patrones pseudo-aleatorios aprendidos en función del tamaño de la red.

Obtener experimentalmente los resultados de la siguiente tabla (los valores de la tabla corresponden a una iteración con actualización sincrónica).

P_{error}	P_{max} / N
0,001	0,105
0,0036	0,138
0,01	0,185
0,05	0,37
0,1	0,61

En este experimento usamos una red de Hopfield '82 para estimar su capacidad de almacenamiento. Para eso generamos patrones pseudo-aleatorios y medimos hasta cuántos se pueden guardar antes de superar un cierto nivel de error P_{error} .

La elección de los parámetros es la siguiente:

- El parámetro n fija el tamaño de la red. Tomé como valor $n=10$, se obtuvieron $N=100$ neuronas. Esto significa que cada patrón es un vector de 100 bits en $(-1;1)$. Probé con un valor de n mayor y el programa se quedaba colgado.
- El parámetro `trials` indica cuántas veces repetimos el experimento para el mismo umbral de error. Cada repetición genera patrones aleatorios distintos, por lo que los resultados pueden variar. Se repite muchas veces y se promedian los resultados para llegar a estimaciones mas estables de la capacidad de la red.
- El parámetro `seed` controla la aleatoriedad. Si no lo fijamos, cada corrida puede dar resultados diferentes porque los patrones iniciales cambian.

Los cambios implementados en las funciones usadas en la parte 1 son las siguientes:

- Patrones (pseudo-aleatorios en vez de imágenes).
- Regla de pesos (dividir por N en vez de P).
- Evaluación (una pasada sincrónica + tasa de error en bits, en vez de convergencia).

Los resultados obtenidos con los parámetros seteados como se muestran a continuación son los siguientes:

- `n=10`
- `trials=100`
- `seed=42`

P_{error}	P_{max} / N
0,001	0.1155
0,0036	0.1410
0,01	0.1840
0,05	0.3770
0,1	0.6140

Estos resultados muestran valores bastantes cercanos a los que se muestran en la tabla inicial, por lo que se puede concluir que aunque se tenga una red con pocas neuronas los resultados siguen a la teoría.

b- Proponga una manera de generar patrones con distintos grados de correlación.

Utilice el método propuesto para analizar cómo varía la capacidad de la red de Hopfield en función de la correlación entre patrones.

Método propuesto

El método propuesto tiene como inicio un patrón base aleatorio de longitud N con valores $(-1,1)$.

Para cada nuevo patrón, se recorre cada posición del vector y se decide, con una probabilidad `proba`, si ese bit se invierte de signo o se deja igual.

Así obtenemos un conjunto de patrones donde todos están correlacionados con el patrón base, y el grado de correlación está dado por `proba`

- Cuanto menor sea el valor del parámetro `proba` más se van a parecer los patrones entre si. (alta correlación)
- Cuanto mayor sea el valor del parámetro `proba` mas diferentes van a ser los patrones.

```
def generar_patrones_correlados(P, N, proba=0.1,
                                rng=np.random.default_rng()):
    """
    Genera P patrones de longitud N con correlación controlada.
    proba: probabilidad de cambiar cada bit respecto al patrón base.
    """
    base = rng.choice([-1, 1], size=N) # patrón base aleatorio
```

```

patterns = [base]
for _ in range(P-1):
    mask = rng.random(N) < proba
    nuevo = base.copy()
    nuevo[mask] *= -1
    patterns.append(nuevo)
return np.array(patterns)

```

Usando las mismas funciones que se usaron en el punto 2a para calcular la capacidad de la red, se arma una tabla para analizar cómo varía la capacidad de la red de Hopfield en función de la correlación entre patrones. Los valores se obtienen con una red de $N=1000$ (mil neuronas)

proba	P_{\max} / N
0.00	0.099
0.05	0.002
0.10	0.002
0.20	0.002
0.50	0.099

Se puede observar que la capacidad de la red disminuye cuando los patrones presentan alta correlación entre sí, es decir, cuando se aumenta el valor de proba.

Lo único que no termino de interpretar es porque con proba=0.50 pega un salto al valor 0.099.

Trabajo Práctico - extra

Implemente una red de Hopfield '82 que aprenda patrones pseudo-aleatorios y estudie qué sucede con los patrones aprendidos cuando algunas interconexiones son eliminadas al azar.

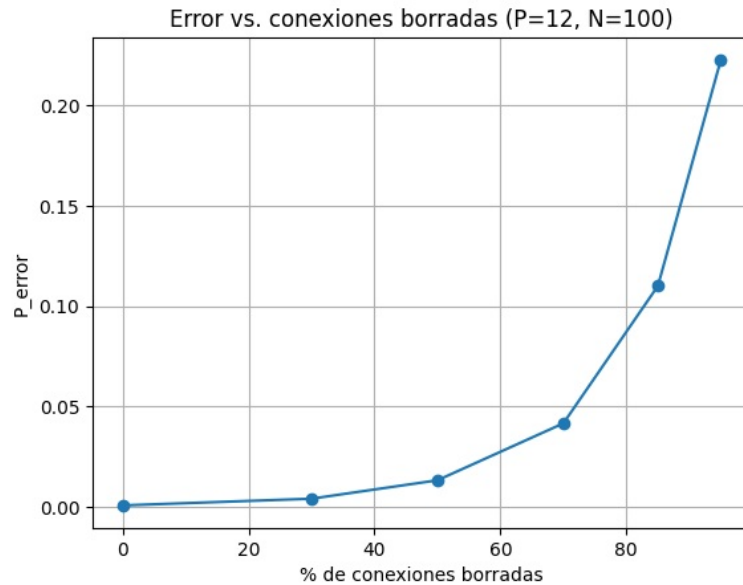
Usando las funciones ya implementadas para toda la parte 2 anterior, la idea práctica consta de los siguientes pasos:

1. Generar un conjunto de patrones pseudoaleatorios.
2. Calcular la matriz de pesos W con la función de Hopfield.
3. Eliminar al azar un porcentaje de interconexiones (poner en cero algunos pesos de W).
4. Ver si los patrones siguen siendo estables (se recuperan después de una pasada sincrónica).

1. ¿Cómo cambia el error en función del porcentaje de sinapsis eliminadas?

Con 100 neuronas y 12 patrones diferentes que se están almacenando en la red, se puede ver en el gráfico que a mayor porcentaje de sinapsis borrada mayor es el error.

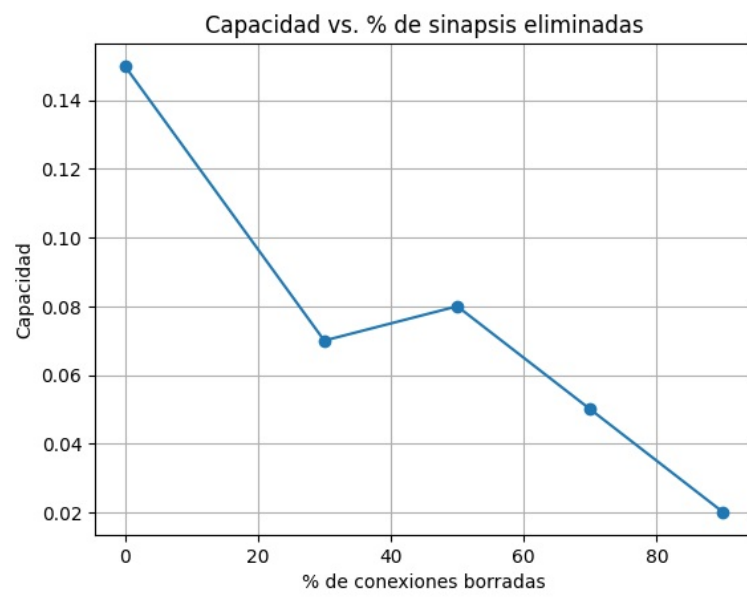
Entre un 60% - 70% de sinapsis eliminada el error está muy cerca de cero, pero cuando se borra un porcentaje mayor al 70%, el error crece casi de forma exponencial.



Error vs sinápsis borrada

2. ¿Cómo cambia la capacidad en función del porcentaje de sinapsis eliminadas?

Con 100 neuronas y 12 patrones diferentes que se están almacenando en la red, se puede ver en el gráfico que, en líneas generales, a mayor porcentaje de sinapsis borrada la capacidad de la red disminuye.



Capacidad vs sinápsis borrada