



**FACULTAD  
DE INGENIERIA**

---

Universidad de Buenos Aires

86.54/66.63 - Redes Neuronales

GUÍA DE EJERCICIOS N°1

Rossi, Francisco 99540 [frrossi@fi.uba.ar](mailto:frrossi@fi.uba.ar)

19 de Octubre de 2020

# Índice

<b>1. Ejercicio 1: Implementación de una Red de Hopfield</b>	<b>2</b>
1.1. Enunciado: . . . . .	2
1.1.a. Implementación: . . . . .	2
1.1.b. Imágenes con elementos agregados . . . . .	2
1.1.c. Imágenes con elementos eliminados . . . . .	3
1.1.d. Imágenes con ruido . . . . .	4
1.1.e. Estados Espurios de la red . . . . .	5
<b>2. Ejercicio 2: Capacidad de la red</b>	<b>6</b>
2.1. Enunciado: . . . . .	6
2.1.a. Implementación: . . . . .	6
<b>3. Ejercicio 3: Robustez</b>	<b>8</b>
3.1. Enunciado: . . . . .	8
3.1.a. Implementación: . . . . .	8
<b>4. Ejercicio 4: Ising</b>	<b>10</b>
4.1. Enunciado: . . . . .	10
4.1.a. Implementación 2-D . . . . .	10
4.1.b. Implementación 1-D . . . . .	11

## 1. Ejercicio 1: Implementación de una Red de Hopfield

### 1.1. Enunciado:

Implemente una red de Hopfield '82 que aprenda imágenes binarias a elección. Pruebe el funcionamiento de la red utilizando distintos patrones de entrada como, por ejemplo, las mismas imágenes con ruido o con elementos borrados o agregados. Verifique la existencia de estados espurios en la red. En particular, verifique si los patrones inversos y la mezcla de los tres patrones son estados espurios (Ver Spurious States, en la sección 2.2, Hertz, Krogh & Palmer, pág. 24).

#### 1.1.a. Implementación:

Para la implementación de la red se define primero la cantidad de neuronas, para esto se tuvo en cuenta que se quieren almacenar imágenes con resolución 50x45 de manera que necesitamos como mínimo  $N = 2250$  neuronas. Por otro lado se consta de 6 patrones a enseñar a la red  $p = 6$ . Las imágenes se llamarán por su contenido: *Paloma*, *Panda*, *Perro*, *torero*, *V*.

Se define una función signo en `signo.m` de manera que cumpla, la misma se usara en diferentes ejercicios a lo largo de esta y las siguientes guías, como función de activación:

$$\text{signo}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (1)$$

Se adapta cada una de las imágenes a vectores  $P_i$  de  $N \times 1$  y se construye la matriz de pesos sinápticos de la red  $W$  como:

$$W = \sum_{i=1}^6 P_i P_i^T \quad (2)$$

#### 1.1.b. Imágenes con elementos agregados

A continuación se realizó la siguiente modificación a la imagen del panda agregandole una flecha y un bigote, de esta manera:

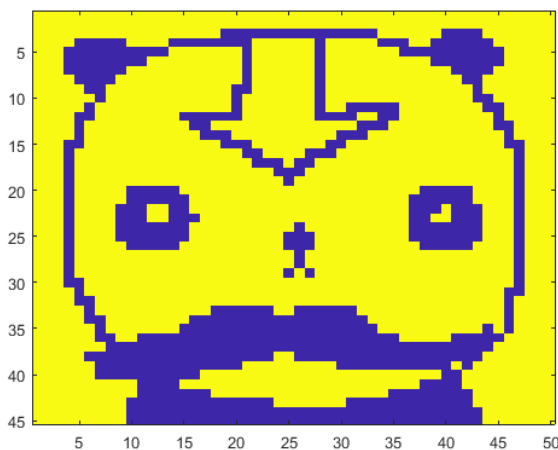


Figura 1: Imagen del panda modificado

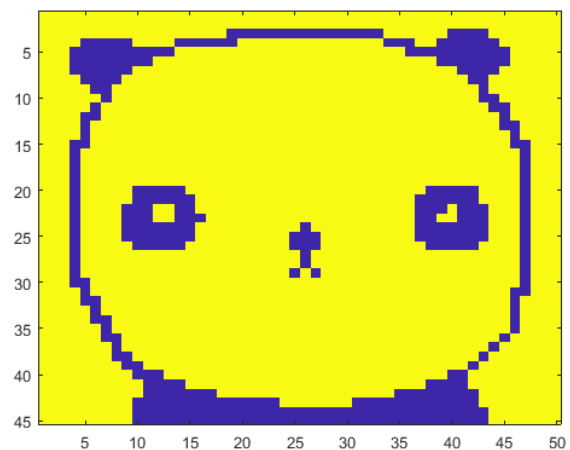


Figura 2: Imagen final a la que convergió el sistema

Se utiliza esta imagen modificada de manera que probar que el sistema posee una capacidad de corregir errores a la entrada y corroboramos efectivamente que el panda es un estado de la red, un mínimo local.

En un principio al realizar esta prueba el sistema terminaba en un estado espurio, pero, eliminando la paloma de la memoria se logró que el sistema funcione correctamente de ahora en adelante la paloma será eliminada del entrenamiento de la red, solo constaremos entonces de 5 patrones. El efecto de la paloma sobre la red puede estar dado por la alta correlación

entre la paloma, el panda y el perro.

### 1.1.c. Imágenes con elementos eliminados

En la siguiente experiencia se le impone a la red un estado inicial de una imagen aprendida pero incompleta, en este caso con 2/3 correcto y otro 1/3 eliminado, como se ve en el caso de  $V$  este converge al estado esperado, no es así en todos los casos como en el caso del panda en el cual termina en un estado espurio cercano al panda.

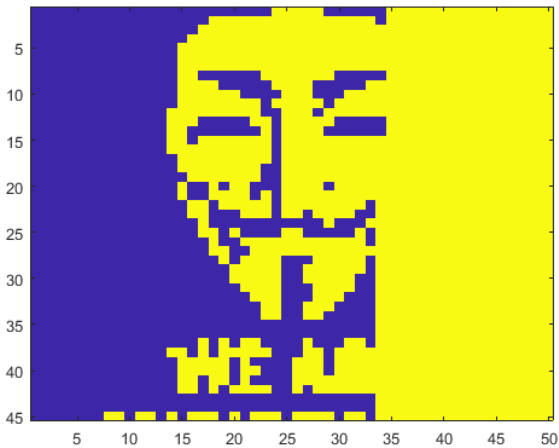


Figura 3: Imagen de  $V$  eliminando un tercio de la imagen

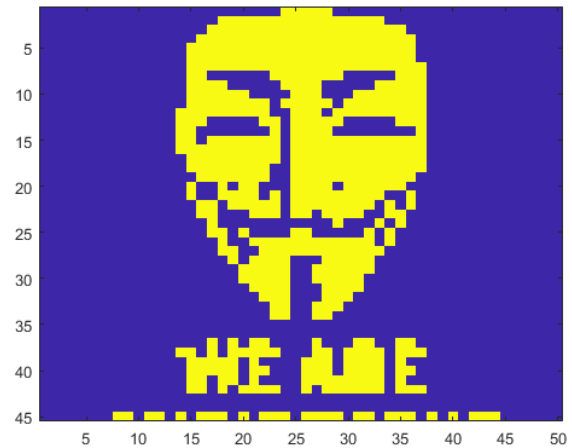


Figura 4: Imagen final a la que convergió el sistema

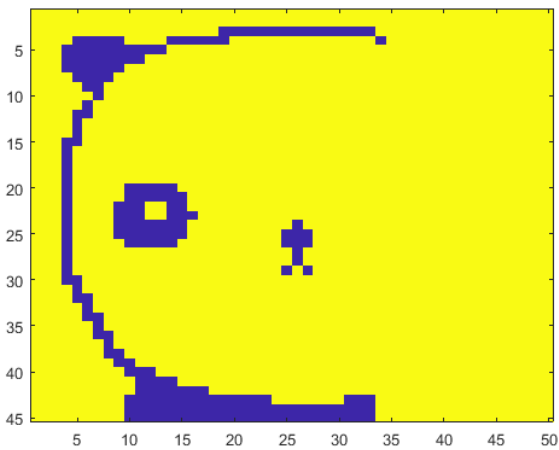


Figura 5: Imagen del panda eliminando un tercio de la imagen

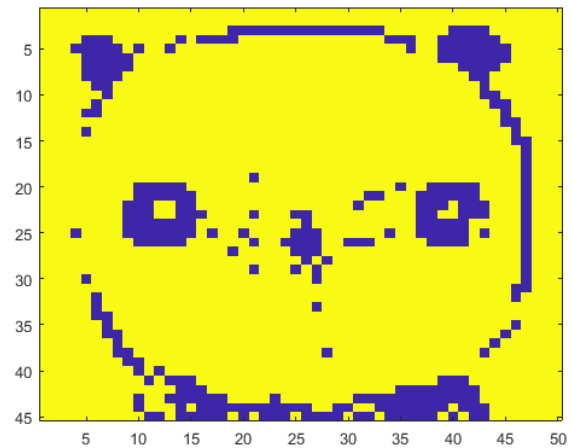


Figura 6: Imagen final a la que convergió el sistema

### 1.1.d. Imágenes con ruido

A continuación se compone una entrada  $h$  de una las 5 imágenes almacenadas sumada a un vector de ruido gaussiano ( $\mathcal{N}(0, 1)$ ) como se ve en los próximos ejemplos todos convergen al estado estable esperado.

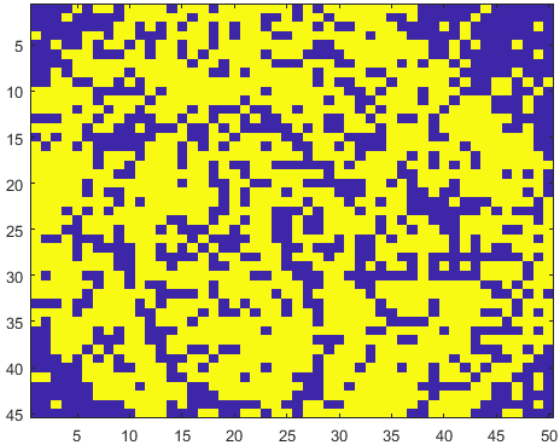


Figura 7: Imagen del perro agregando ruido gaussiano

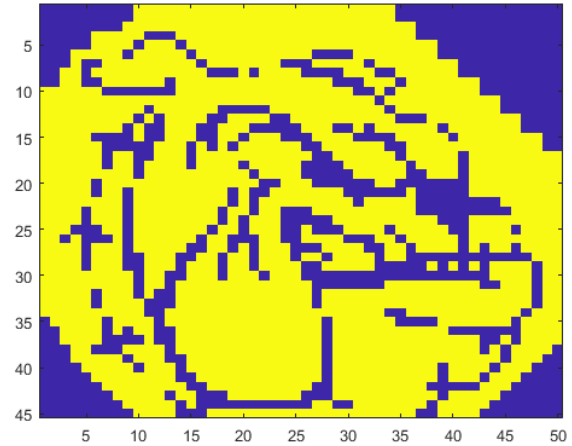


Figura 8: Imagen final a la que convergió el sistema

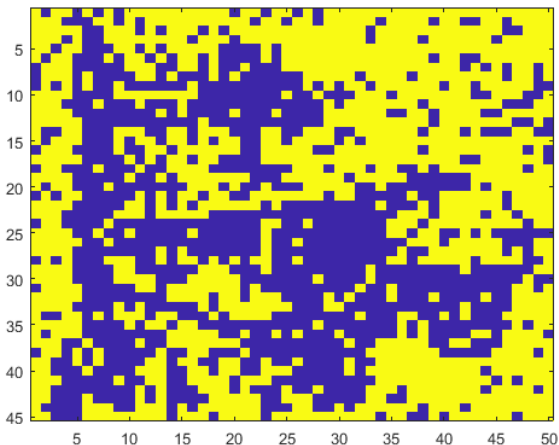


Figura 9: Imagen del quijote agregando ruido gaussiano

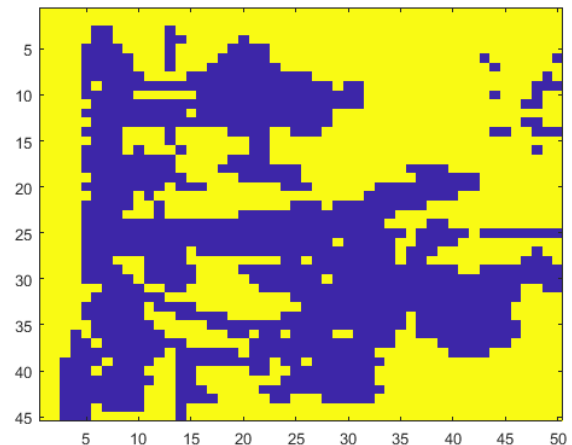


Figura 10: Imagen final a la que convergió el sistema

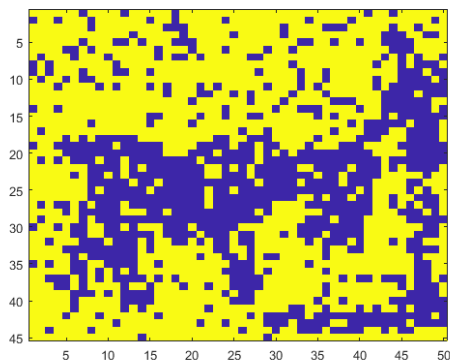


Figura 11: Imagen del torero agregando ruido gaussiano

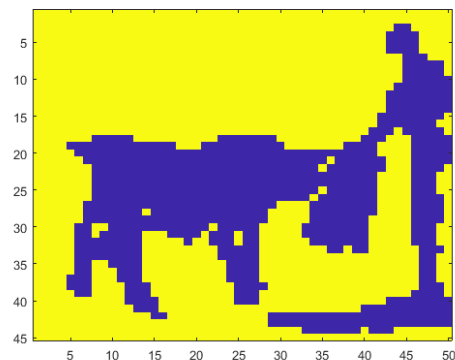


Figura 12: Imagen final a la que convergió el sistema

### 1.1.e. Estados Espurios de la red

#### Estados negados

En una red de Hopfield los estados negados tienen la misma energía que los no negados de manera que también serán un estado estable, ya que los estados estables de la red de Hopfield están dados por mínimos de energía. Esto se comprobó con todas las imágenes, debajo se muestra el caso del panda y del torero al darle esta condición inicial la red no se mueve de este estado.

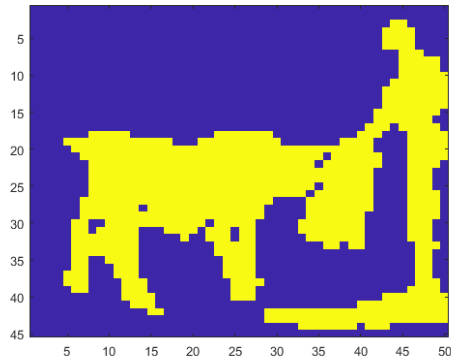


Figura 13: Imagen del torero invertido

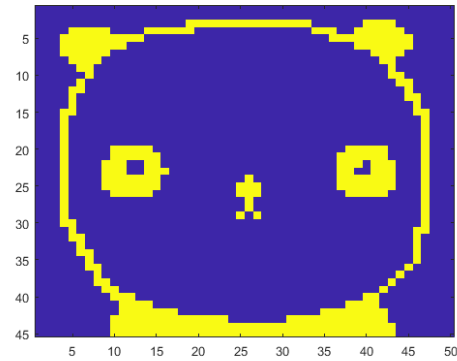


Figura 14: Imagen del panda invertido

#### Estado Mezcla

Teóricamente la mezcla de estados estables es un estado estable, se probó sumando algebraicamente los vectores de los cinco patrones pero este no es un estado estable, probablemente por la alta correlación entre el panda y el torero, reduciendo el número de patrones a tres se logró comprobar la existencia de un estado estable dado por una mezcla de estos patrones, a continuación se muestra el caso de Quijote + Torero + V.

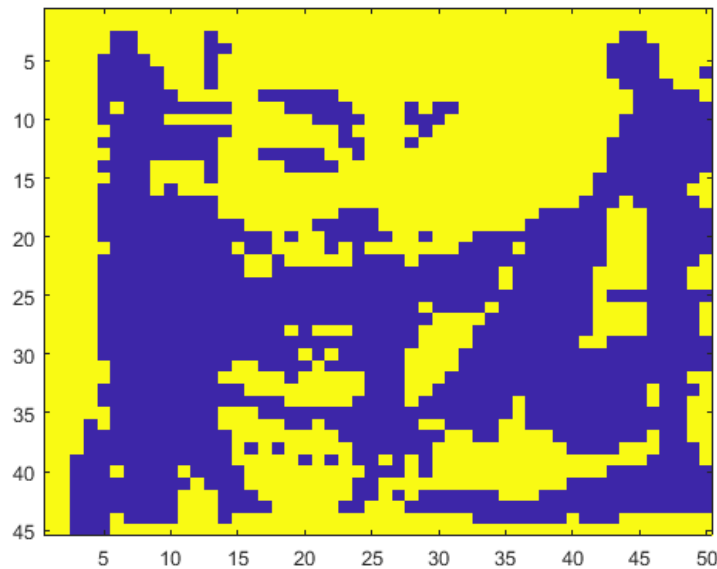


Figura 15: Estado mezcla

## 2. Ejercicio 2: Capacidad de la red

### 2.1. Enunciado:

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).

Tabla 1: Tabla la capacidad de la red en función de la probabilidad de error.

$\mathbb{P}_{error}$	$p_{max}/N$
0.001	0.105
0.0036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

#### 2.1.a. Implementación:

Para esto se define el error como la media en cantidad de neuronas y cantidad de patrones donde no coincide el patrón enseñado ( $P$ ) y la salida dada una entrada de este patrón aprendido ( $W*P$ ).

$$err = mean(mean((signo(W * P) - P) = 0)) \quad (3)$$

Se realizó para cada valor  $\mathbb{P}_{error}$  el cálculo de la cantidad máxima de patrones  $p_{max}$  para diferentes cantidades de neuronas  $N$  entre 1 a 510, de a paso de 10 en casi todos los casos. Se van incrementando la cantidad de patrones aprendidos, los cuales son pseudo aleatorios, hasta alcanzar o superar el error máximo propuesto, cuando esto sucede se incrementa en *paso* la cantidad de neuronas y se repite hasta alcanzar las 510 neuronas.

Se obtuvieron una serie de valores de capacidad para cada cantidad de neuronas para cada valor de  $\mathbb{P}_{error}$ , a partir estos valores de capacidad (uno por cada cantidad de neuronas) se obtuvieron los valores medios de las capacidades para cada probabilidad de error.

En la **Fig. 16** se muestra para cada  $\mathbb{P}_{error}$  la serie de valores de  $p_{max}$  para cada valor de  $N$ .

La pendiente de cada una de estas rectas es lo que buscamos obtener:  $C = \frac{p_{max}}{N}$  es la capacidad de la red, luego calculamos la media de este valor a lo largo de cada punto de cada recta, para obtener la capacidad media en cada caso.

Los resultados obtenidos se reflejan en la **Tabla 2** con el correspondiente error relativo porcentual  $\Delta \% = \frac{|C_{est} - C_{tabla}|}{C_{tabla}}$

Tabla 2: Tabla la capacidad de la red en función de la probabilidad de error.

$\mathbb{P}_{error}$	$C_{tabla}$ [tabla]	$C_{est}$	$\Delta \%$
0.001	0.105	0.1317	25.43
0.0036	0.138	0.1367	0.94
0.01	0.185	0.1799	2.76
0.05	0.37	0.3719	0.51
0.1	0.61	0.5866	3.84

Podemos observar que para todos los casos excepto para  $\mathbb{P}_{error} = 0,001$  los valores tienen un error menor al 4 %. Para este caso simularemos con un paso menor en la cantidad de neuronas, de manera de obtener mas valores de  $p_{max}$  y así mejorar nuestra aproximación.

Antes utilizábamos un vector de  $N \in [1, 510]$  con paso 10, ahora el paso se redujo a 1 obteniéndose:

$$C_{est} = \frac{p_{max}}{N} |_{\mathbb{P}_{error}=0,001} = 0,1126 \implies \Delta \% = 7,24 \quad (4)$$

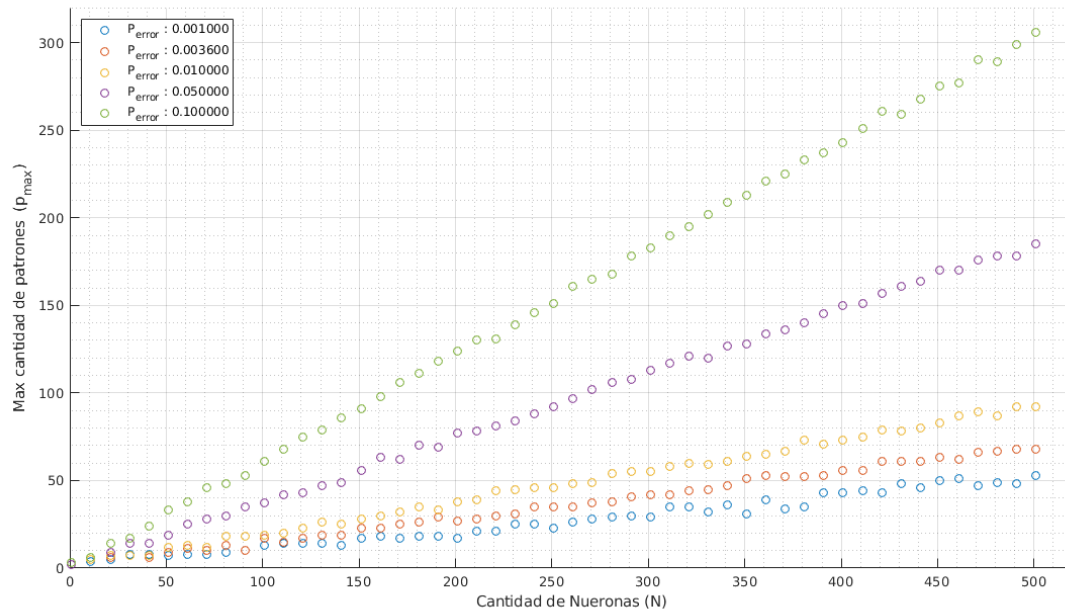


Figura 16: Capacidad de la red

Se redujo un 18,19% el error relativo porcentual y podríamos tener una mejor aproximación aumentando la cantidad de neuronas máxima, así obteniendo mas valores de  $p_{max}$  y reduciendo así la varianza.



### 3. Ejercicio 3: Robustez

#### 3.1. Enunciado:

Implemente una red de Hopfield '82 que aprenda patrones pseudo-aleatorios y muestre qué sucede con los patrones aprendidos cuando algunas interconexiones son eliminadas al azar. Estime cuánto disminuye la capacidad en función de la eliminación de interconexiones.

##### 3.1.a. Implementación:

Para analizar el deterioro de la red a medida que algunas interconexiones son eliminadas al azar se repite **samples** veces (10) para  $p = 10$  y  $p = 15$  patrones y  $N = 100$  neuronas el siguiente algoritmo:

- Construyo una matriz de pesos sinápticos  $W$  a partir de  $p$  patrones aleatorios
- Se eliminan un  $x\%$  desde 0 a 100 % de interconexiones (elegidas al azar) yendo de a pasos de a 5 %
- Se calcula el error  $err$

Nuevamente utilizaremos la siguiente definición del error.

$$err = mean(mean((signo(W * P) - P) = 0)) \quad (5)$$

Este error se calcula para cada paso de los deterioros y se obtiene el gráfico del error en función de la cantidad de sinapsis eliminadas que se muestra en la **Fig. 17**.

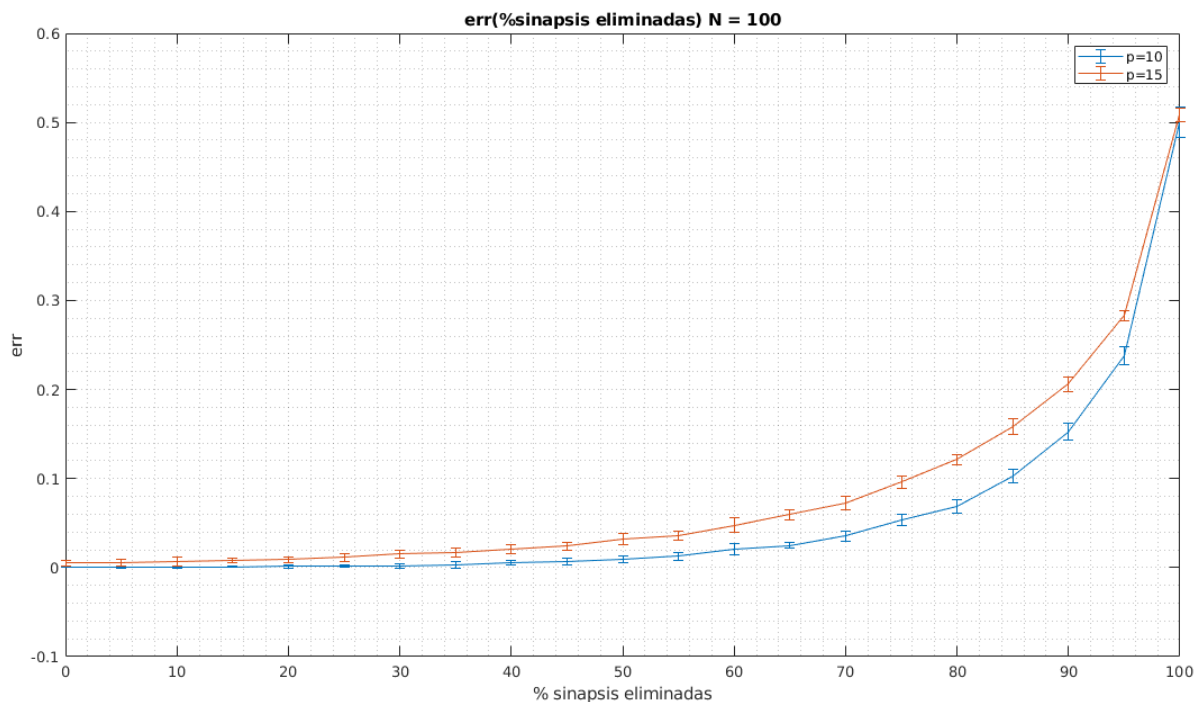


Figura 17: Error en función de la cantidad de sinapsis eliminadas

El gráfico **Fig. 17** demuestra la robustez de la red de manera que para un % de hasta un 75 % el error no supera el 10 %.

Como vemos cuando la cantidad de sinapsis eliminada llega al 100 % el error llega a su máximo del 50 %. Debido a que la función de activación de las neuronas es la función signo, y por ende, la salida de cada neurona tiene dos valores posibles: -1 o 1. Dada una matriz  $W$  nula (se eliminaron todas las sinapsis), el valor de la salida será siempre un vector de unos. Además

dado que los patrones enseñados son generados a partir de binomiales  $-1, 1$  con probabilidad  $\mathbb{P} = 0,5$ , la probabilidad de error máxima es efectivamente la misma que la probabilidad de cada bernoulli de ser 1, en este caso 0,5.

Luego se quiere evaluar cuanto disminuye la capacidad de la red en función del porcentaje de sinapsis eliminadas. Para esto se aumenta  $p$  hasta que el error de la red sea menor al  $p_{error} = 0,01$ , idéntico a lo realizado en el ejercicio 2. Luego se almacena este valor para el porcentaje correspondiente  $x\%$  y se aumenta la cantidad de sinapsis eliminadas. Se repite el proceso hasta el 100% y se calcula la capacidad en cada caso como  $p/N$ . Este proceso se repite **samples** veces (1) para obtener el gráfico de la **Fig. 18** con el valor medio de las capacidades.

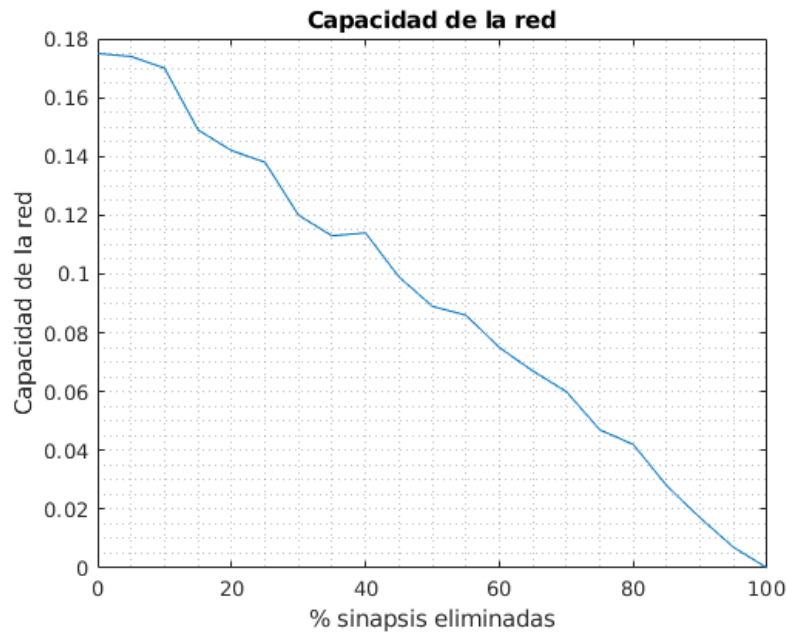


Figura 18: Capacidad en función de la cantidad de sinapsis eliminadas

El gráfico **Fig. 18** muestra como la capacidad disminuye a medida que la cantidad de sinapsis eliminadas aumenta. Iniciando en  $mean(C) = 0,175$  es decir  $p_{max} = 0,175 * N_{neu}$ , hasta cero. El gráfico muestra una cierta linealidad en la relación entre la cantidad de sinapsis eliminada y la capacidad de la red.

## 4. Ejercicio 4: Ising

### 4.1. Enunciado:

Simular un modelo de Ising en una y dos dimensiones. Encontrar la temperatura crítica para ambos casos.

#### 4.1.a. Implementación 2-D

Para simular este modelo se determina la matriz  $W$  de manera de cumplir con lo indicado en el siguiente dibujo:

Se inicializa el vector  $S$  de manera aleatoria con probabilidad  $1/2$  ya que estamos en temperaturas altas. Se calcula la energía inicial  $H$ .

Luego para temperaturas entre 4 y 0.5 (se hayo empíricamente que para este rango se llega a apreciar la sigmoidea desde spines no alineados hasta totalmente alineados. Se repite `samples` veces (35) el siguiente proceso:

Se recorren aleatoriamente cada uno de los spines y se invierte su valor, si la energía disminuye o permanece constante se acepta el cambio, si la energía en cambio aumenta se acepta el cambio con probabilidad  $p = e^{-\frac{\Delta H}{kT}}$ , luego se calcula la media del vector  $S$  donde se espera que pase de valer 0 (se inicializo de manera aleatoria con  $p = 0,5$ , donde todos los dipolos no están alineados a 1 o -1 donde todos los dipolos están alineados.

Se ajustan los puntos la función `sigm_fit` y se obtiene el gráfico de la **Fig. 19**.

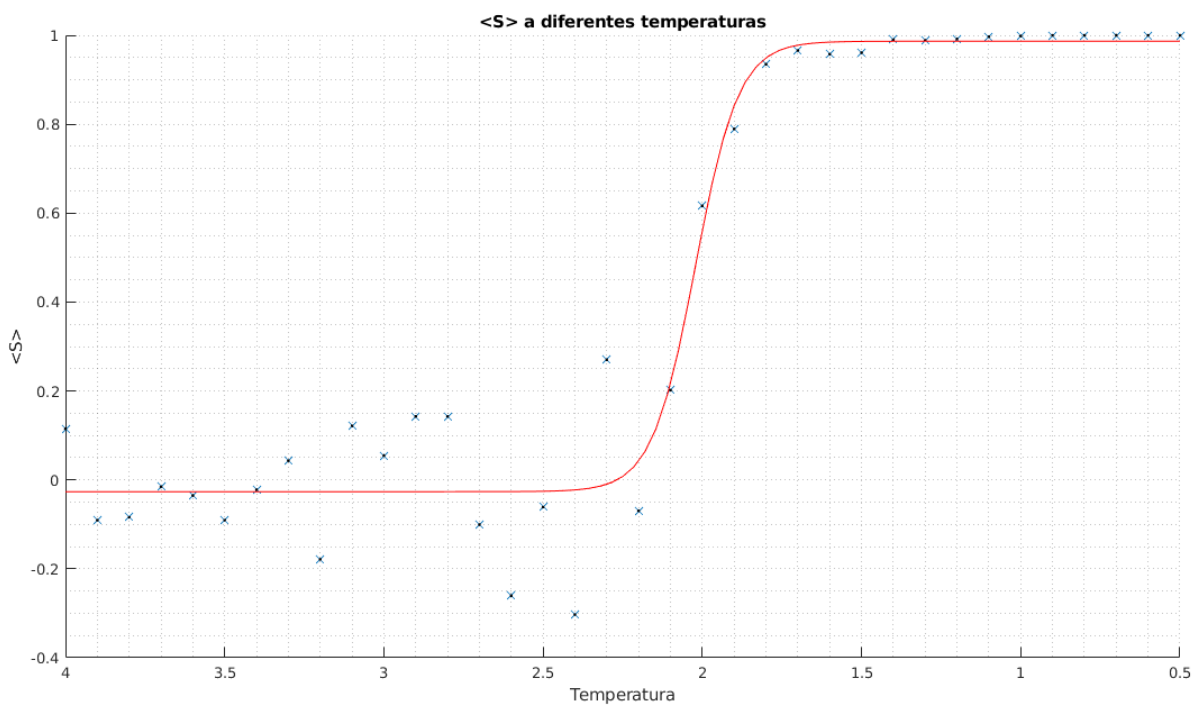


Figura 19: campo medio en función de la temperatura caso 2D

Cuanto mayor es la temperatura mayor es la probabilidad de aceptar un cambio que aumente la energía, es por esto que a temperaturas altas no se logra la alineación de los spines con el método de gradiente descendiente. Al bajar la temperatura la probabilidad de este fenómeno es cada vez menor, y esta demostrada la existencia de la temperatura crítica la cual determina el punto de inflexión a partir de la cuál la alineación total de los spines comienza a tener probabilidad alta. Se observa en el gráfico que la temperatura crítica es aproximadamente 2 para un  $k = 1$ .

#### 4.1.b. Implementación 1-D

La implementación de este caso es idéntica al caso bidimensional a excepción de la matriz de pesos sinápticos. En la anterior se construye a partir de la relación con vecinos tanto al norte, sur, este y oeste, en este caso solo tenemos un valor de peso sináptico diferente de cero si la distancia entre ambos spines es 1.

En la **Fig. 20** se muestra que no existe la temperatura crítica en el caso unidimensional, se trata de ajustar con la función `sigm_fit` pero no es posible. No hay temperatura crítica.

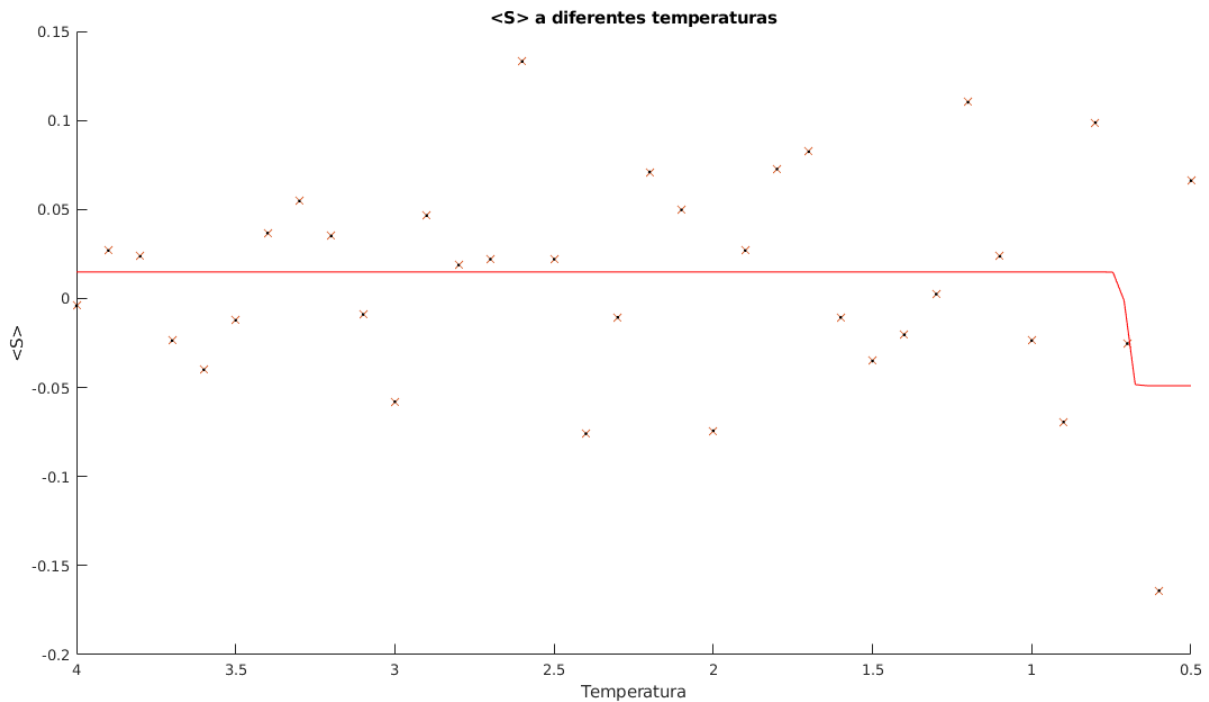


Figura 20: campo medio en función de la temperatura caso 1D