

Segundo Trabajo Práctico

Sistemas

Adaptativos: redes

neuronales

David Alejandro Jorge Tasé

Ejercicio 1

- a) Para la implementación de la red se define primero la cantidad de neuronas, para esto se tuvo en cuenta que se quieren almacenar patrones con resoluciones 45x60 y de 50x50, 3 imágenes para ambos casos, por lo que para un rendimiento óptimo se implementan dos redes, una de 2700 neuronas y otra de 2500.

Se adapta cada una de las imágenes a vectores columnas P_i y P_j se construye la matriz de pesos sinápticos de la red como:

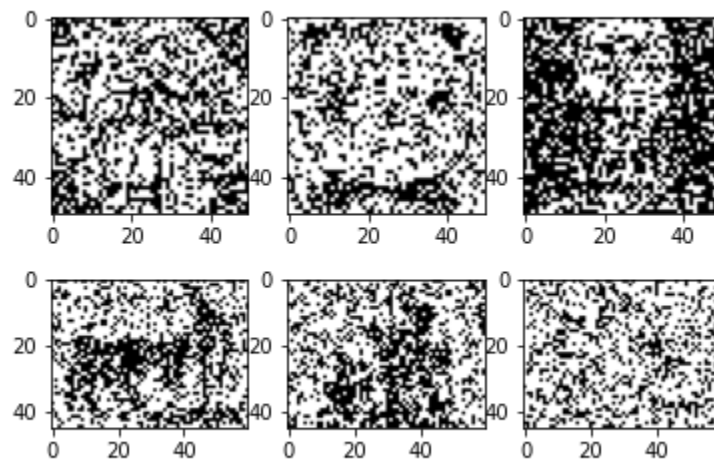
$$w_1 = \sum_{i=1}^3 P_i P_i^T$$

$$w_2 = \sum_{j=1}^3 P_j P_j^T$$

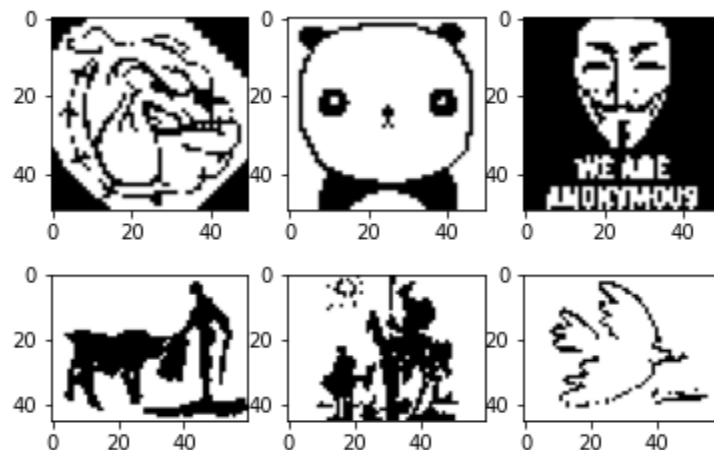
A w_1 se le enseñaron las imágenes correspondientes a Perro, Panda y V, en tanto que a w_2 se le asignaron Quijote, Torero, y Paloma. Para comprobar el buen funcionamiento de las redes se les pasa las imágenes correspondientes y se les hace una actualización asincrónica tras lo cual las imágenes deben verse intactas.

- b) A continuación se evalúa la evolución de la red al presentar versiones alteradas de las imágenes aprendidas: agregado de ruido, elementos borrados o agregados.

- Con ruido agregado:

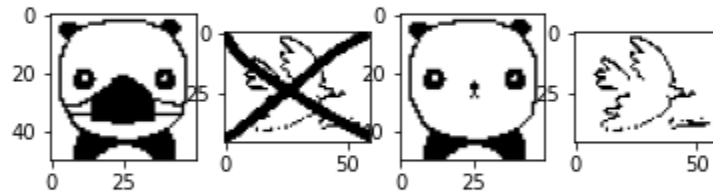


Imágenes con 1000 píxeles ruidosos

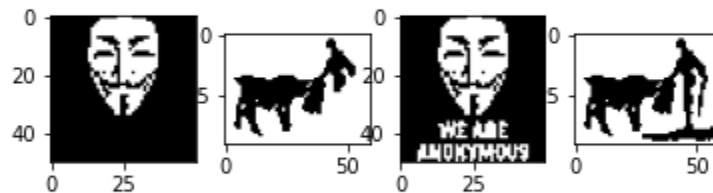


Imágenes después de ser limpiadas por las redes

- Con elementos agregados:



- Con elementos borrados:

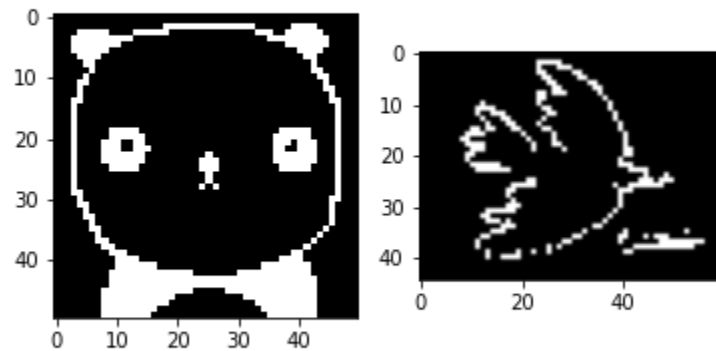


De esta forma se comprueba que las imágenes enseñadas representan mínimos de energía de la red.

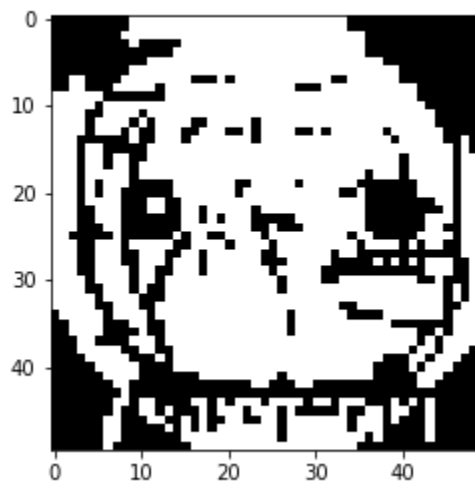
c) Estados espurios de la red

- Estados negativos: En una red de Hopfield los estados negativos tienen la misma energía que los no negativos de manera que también serán un estado estable, ya que están dados por mínimos de energía. Esto se

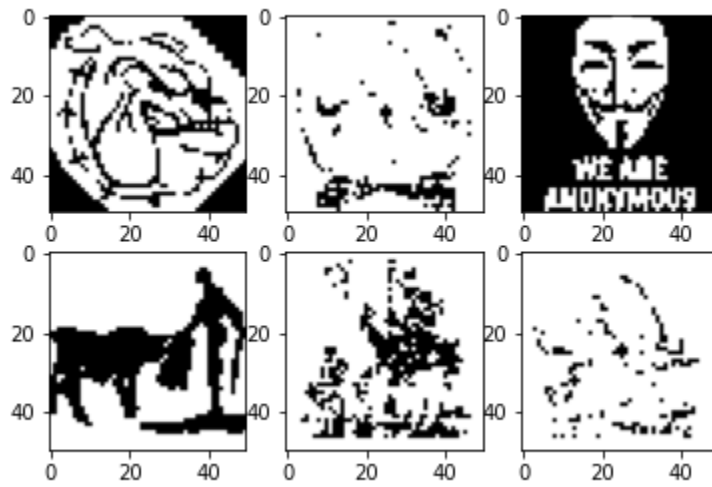
comprobó con todas las imágenes, debajo se muestra el caso del Panda y la Paloma, al darle esta condición inicial la red no se mueve de este estado.



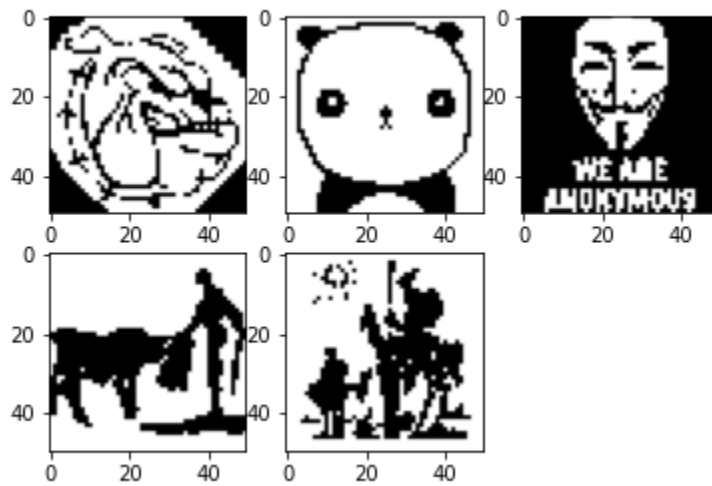
- Estados combinados: la mezcla de estados estables es un estado estable. Se logró comprobar la existencia de un estado estable dado por una mezcla de estos patrones, a continuación se muestra el caso de Perro + Panda + V.



- d) Entrenamiento con las 6 imágenes para una sola red: para esto se reconfiguraron las imágenes de 45 x 60 píxeles a 50 x 50 y se creó una nueva red con conexiones w . A continuación se comprueba si las imágenes fueron aprendidas o no por la red:



Al parecer, la red no fue capaz de aprender todos los patrones correctamente, esto se debe a la capacidad de almacenamiento de la red y su probabilidad de error en función del número de patrones. Sin embargo si solo le enseñamos 5 patrones a la red es capaz de almacenarlos en su totalidad.



Ejercicio 2

- Se comienza definiendo las principales variables a tener en cuenta: el vector de probabilidad de error esperado ($e_{probabilidad}$), el calculado (p_{error}), y las dimensiones

de la imagen (110 x 110 para este caso) dada por una distribución de Bernoulli con valores binarios de probabilidad de ocurrencia aproximadamente 0.5. Luego se implementa el algoritmo correspondiente a una red de Hopfield con actualización sincrónica. Obteniéndose los siguientes resultados:

p_{error}	$\frac{p_{max}}{N}$ (esperado)	$\frac{p_{max}}{N}$ (calculado)	Error relativo %
0.001	0.105	0.09636	8.23
0.0036	0.138	0.123	10.87
0.01	0.185	0.1734	6.27
0.05	0.37	0.3542	4.27
0.1	0.61	0.5853	4.05

b) En este caso se probaron dos formas de obtener la capacidad de la red de Hopfield cuando existe gran similitud entre los patrones que se le enseñan:

variante 1: se sigue una lógica ligeramente diferente a la implementada en el inciso a):

1. Se crea la red de Hopfield y se le enseña un primer patrón en forma de vector columna (10000, 1).
2. Se modifica aleatoriamente el valor de 10 posiciones (o píxeles) del primer patrón creando de esta manera uno nuevo muy parecido al primero.
3. Se le enseña este último patrón a la red de Hopfield y se calcula la probabilidad de error y la capacidad.
4. Se repiten los pasos del 1 al 3 de tal manera que en cada ciclo se le enseña un nuevo patrón con solo 10 valores distintos al anterior y almacenando la capacidad calculada para las probabilidades de error aproximadas a (0.001, 0.0036, 0.01, 0.05, 0.1). Obteniendo la siguiente tabla:

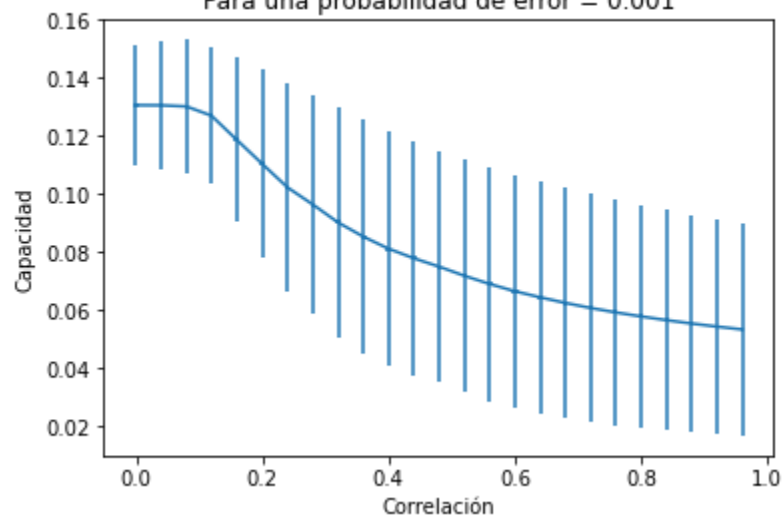
p_{error}	$\frac{p_{max}}{N}$
-------------	---------------------

0.0012	0.0004
0.0036	0.0016
0.0101	0.0038
0.0501	0.0231
0.1002	0.0567

Se puede ver claramente que para patrones similares la capacidad de la red disminuye considerablemente.

variante 2: una forma de medir la similitud entre dos variables es la correlación. En este caso, queremos observar cómo se comporta la capacidad de la red de Hopfield al aprender patrones que están correlacionados. El coeficiente de correlación varía entre 1 y -1, donde 1 indica una correlación positiva perfecta y -1 indica una correlación negativa perfecta. El siguiente gráfico da una idea de la relación entre la capacidad y la correlación:

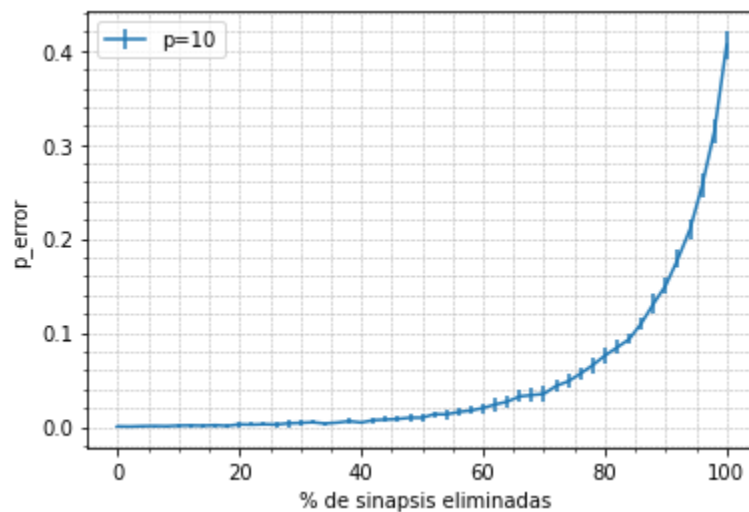
Capacidad de la red en función de correlación entre los patrones aprendidos
Para una probabilidad de error = 0.001



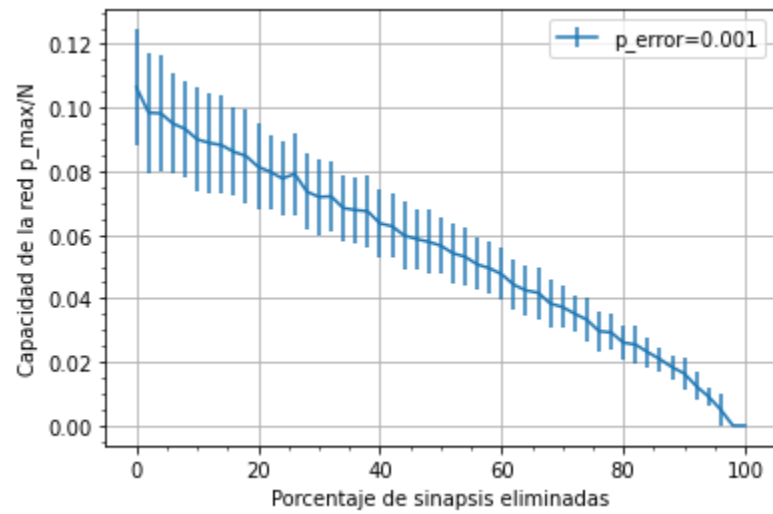
Evidentemente, la capacidad de la red de Hopfield se reduce a medida que aprende patrones que tienen una mayor correlación entre ellos.

Ejercicio 3

- a) Para cuantificar el deterioro de la red en cuanto a probabilidad de error (p_{error}) y porcentaje de sinapsis eliminadas: se construyó una red de 100 neuronas a partir de 10 patrones pseudoaleatorios, y se eliminaron conexiones progresivamente con un incremento del 5 % en cada iteración para el cálculo del p_{error} , y todo este proceso se repite 10 veces para cada valor de %.

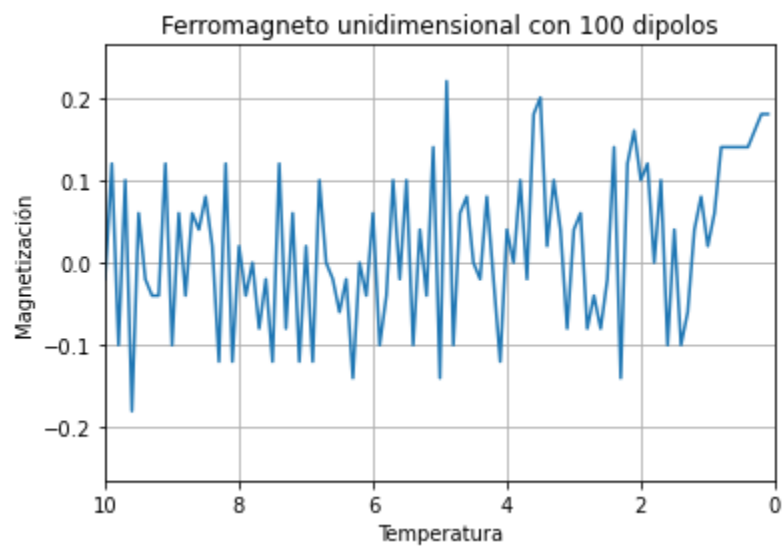


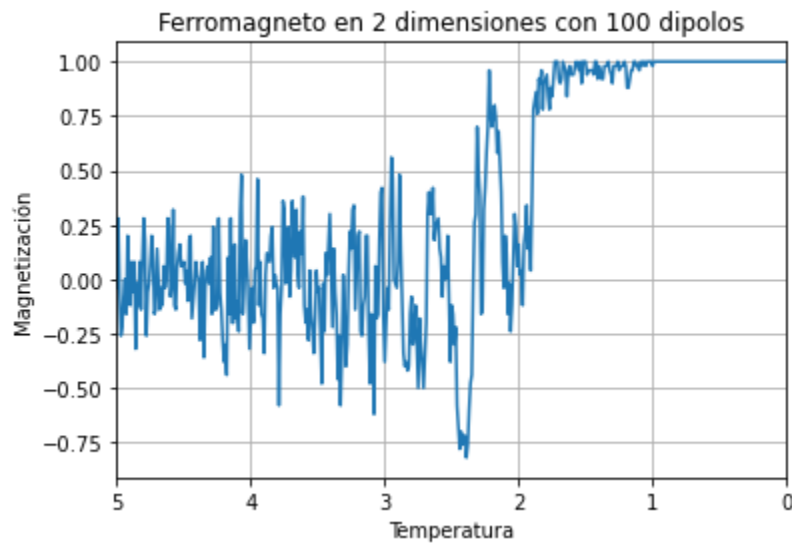
- b) Para este inciso se ejecuta un algoritmo similar, mostrando en este caso la capacidad de la red:



Ejercicio 4

Se implementa el modelo de Ising unidimensional y bidimensional siguiendo el algoritmo descrito en clase.





Para el caso unidimensional se comprueba que no se alcanza nunca la magnetización. Sin embargo para el caso bidimensional esta se alcanza rápidamente luego de superar los 2 grados con un factor $B = 1$.

Ejercicio 5

Se implementa un perceptrón de dos capas (4 neuronas en la primera y 1 en la segunda) utilizando el algoritmo de “*Simulated Annealing*”, se seleccionaron los hiperparámetros con los siguientes valores:

- $\alpha = 0.99$ factor de enfriamiento
- $\mu = 0$ media y varianza para la actualización de los pesos de las neuronas
- $\text{var} = 0.8$ se seleccionó una varianza alta ya que se comprobó que ayuda a reducir el número de iteraciones
- $B = 1$
- $T_{\text{max}} = 20$ temperatura máxima
- $T_{\text{min}} = 1e-6$ temperatura mínima
- $E_{\text{target}} = 1e-4$ error objetivo

