UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

MAESTRÍA EN SIMULACIÓN NUMÉRICA Y CONTROL

TRABAJO PRACTICO 1:

RED DE HOPFIELD 82

Presentada como requisito parcial para la aprobación del curso
Sistemas Adaptativos y Redes Neuronales.

Estudiante:
Fredy Andrés Mercado Navarro
DNI: 94.872.342

Profesor titular:
Sergio Lew

Clases Prácticas:
Ana Laura Vadnjal

17 de octubre de 2013
Buenos Aires, Argentina

Resumen

Este informe contiene el desarrollo del trabajo práctico número 1 sobre la Red de Hopfield, red neuronal artificial recurrente, inventada por John Hopfield y publicada en enero de 1982, donde expone el surgimiento de propiedades computacionales que fueron llamadas "propiedades emergentes", las cuales se manifiestan en sistemas con un gran número de componentes más simples llamados neuronas. Otro aspecto importante es el concepto de memoria direccionable por contenido, con la cual es posible recuperar un patrón aprendido a partir de partes incompletas del mismo patrón. Se estudia, en resumen, el aprendizaje, la capacidad, y otras propiedades de ésta red.

Índice general

1. Ejercicios teóricos **4**
1.1. Punto 1 4
1.1.1. Literal a 4
1.1.2. Literal b 5
1.2. Punto 2 6
1.2.1. Desarrollo 6

2. Ejercicios Prácticos **7**
2.1. Punto 1 7
2.1.1. Desarrollo 7
2.2. Punto 2 9
2.2.1. Desarrollo 9
2.3. Punto 3 11
2.3.1. Desarrollo 11
2.4. Punto 4 12
2.4.1. Desarrollo 12
2.5. Punto 5 14
2.5.1. Desarrollo 14

3. Comentarios y Conclusiones **19**

Índice de figuras

2.1. Imagenes aprende la red de Hopfield 82. 7
2.2. Evolución de la imagen del torero con ruido. 8
2.3. Evolución de la imagen del torero con ruido 2. 9
2.4. Evolución de la imagen del Quijote con ruido. 9
2.5. Evolución de la imagen del torero sin cabeza, capa ni patas traseras. . . . 10
2.6. Evolución de la imagen inversa de la paloma. 10
2.7. Capacidad de la red de Hopfield 82 para N=400, M=5. 11
2.8. Capacidad de la red de Hopfield 82 para N variable, M=50. 12
2.9. Capacidad de la red de Hopfield en función de la variación de interconexiones eliminadas de $\underline{\underline{W}}$. 13
2.10. Capacidad calculada enseñando solo los patrones inversos. 13
2.11. Bits erroneos al mostrar mezclas aleatorias de 3, 5 y 7 patrones a la red. 14
2.12. Magnetización ferromagnética por espin para una malla 1D de 100 elementos. $i = 10000$. 16
2.13. Estado inicial de magnetización ferromagnética para una mapa de espines de 100 x 100. $i = 0$, $T = 1$. 16
2.14. Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 20$, $T = 1$. 17
2.15. Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 50$, $T = 1$. 17
2.16. Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 200$, $T = 1$. 18
2.17. Magnetización ferromagnética por espin para una mapa de spins de 30 x 30. Barridos $i = 1000$. 18

Índice de cuadros

2.1. Capacidad en función de la cantidad de neuronas N. 11

Capítulo 1

Ejercicios teóricos

1.1. Punto 1

La función de energía para una red de Hopfield 82 se define como:

$$E = -\frac{1}{2}\sum_{ij} W_{ij} s_i s_j$$

1.1.1. Literal a

Demuestre que la función de energía está acotada inferiormente (sugerencia: exprese la función de energía en su forma matricial).

Desarrollo

En otras palabras, se nos pide demostrar que existen patrones que hacen mínimo el cálculo de la energía. La energía de estos patrones es un mínimo local. Al final llegaremos a la conclusión de que esos patrones son los mismos que usamos para entrenar la red. Debemos entonces calcular la energía para un patrón de entrada y demostrar que cualquier patrón distinto de ese me dará una energía mayor.

En forma matricial, la función de energía se expresa como:

$$E = -\frac{1}{2}\ \underline{s}^T \cdot \underline{\underline{W}} \cdot \underline{s} \tag{1.1}$$

Supongamos que entrenamos la red con un patrón $\underline{p}$, así:

$$\underline{\underline{W}} = \underline{p}\underline{p}^T - \underline{\underline{I}}$$

Si le mostramos a la red un patrón y obtenemos la salida $\underline{s}$, la energía será:

$$E = -\frac{1}{2}\ \underline{s}^T \cdot \left(\underline{p}\underline{p}^T - \underline{\underline{I}}\right) \cdot \underline{s}$$

$$E = -\frac{1}{2}\left(\underline{s}^T \cdot \underline{p}\underline{p}^T \cdot \underline{s} - \underline{s}^T \cdot \underline{\underline{I}} \cdot \underline{s}\right)$$

$$E = -\frac{1}{2}\left[\left(\underline{s}^T \cdot \underline{p}\right)\left(\underline{p}^T \cdot \underline{s}\right) - \underline{s}^T \cdot \underline{s}\right]$$

$$E = \frac{1}{2}\left[\underline{s}^T \cdot \underline{s} - \left(\underline{s}^T \cdot \underline{p}\right)\left(\underline{p}^T \cdot \underline{s}\right)\right]$$

Por las propiedades del producto escalar sabemos que $\left(\underline{s}^T \cdot \underline{p}\right) = \left(\underline{p}^T \cdot \underline{s}\right)$, luego:

$$E = \frac{1}{2}\left[\underline{s}^T \cdot \underline{s} - \left(\underline{s}^T \cdot \underline{p}\right)^2\right]$$

Las cantidades $(\underline{s}^T \cdot \underline{s})$ y $(\underline{s}^T \cdot \underline{p})$ son máximas para una salida $\underline{s} = \underline{p}$, lo que indica que la energía para este caso es un mínimo local. La energía mínima local estará dada entonces por:

$$E = \frac{1}{2}\left[\underline{p}^T \cdot \underline{p} - \left(\underline{p}^T \cdot \underline{p}\right)^2\right]$$

Si hacemos $n = \underline{p}^T \cdot \underline{p}$ tendremos:

$$E_{min} = \frac{1}{2} n\left[1 - n\right]$$

Lo que demuestra que la función de energía está acotada inferiormente, limitada por el número de elementos diferentes de cero que posea el patrón enseñado p.

1.1.2. Literal b

Demuestre que si se utiliza la regla dinámica

$$s_i = sign\left(\sum_j W_{ij} s_j\right)$$

para actualizar el estado de la red, la energía siempre decrece o permanece igual.

Desarrollo

Recordemos que la función de energía para una red de Hopfield 82 se define como:

$$E(\underline{s}) = -\frac{1}{2}\sum_{ij} W_{ij} s_i s_j$$

Asumiremos que W_{ij} es simétrica. Ahora supongamos que actualizamos una sola neurona s_k. Esta neurona conservará el mismo valor o lo cambiará. Demostraremos entonces que la energía decrece o permanece igual actualizando la salida desde un estado s_i hasta un estado s_i'. Sea E la energía antes de la actualización y E' la energía después:

$$E = -\frac{1}{2}\sum_{ij} W_{ij} s_i s_j = -\frac{1}{2}\sum_{ij, i\neq k, j\neq k} W_{ij} s_i s_j - \frac{1}{2}\sum_i W_{ik} s_i s_k - \frac{1}{2}\sum_j W_{kj} s_k s_j$$

Con lo anterior solo reescribimos la sumatoria para poner por separado los términos que dependen de s_k. Ahora calculamos la energía para E':

$$E' = -\frac{1}{2}\sum_{ij} W_{ij} s_i s_j = -\frac{1}{2}\sum_{ij, i\neq k, j\neq k} W_{ij} s_i s_j - \frac{1}{2}\sum_i W_{ik} s_i s_k' - \frac{1}{2}\sum_j W_{kj} s_k' s_j$$

Ahora calculamos:

$$\Delta E = E' - E$$

$$\Delta E = -\frac{1}{2}\sum_j W_{kj} s'_k s_j - \frac{1}{2}\sum_i W_{ik} s_i s'_k + \frac{1}{2}\sum_j W_{kj} s_k s_j + \frac{1}{2}\sum_i W_{ik} s_i s_k$$

Teniendo en cuenta la simetría de la matriz de pesos sinápticos obtenemos:

$$\Delta E = \sum_i W_{ki} s_i \left(s_k - s'_k\right)$$

En este punto existen dos posibilidades para una actualización:

$$s_k = -1 \rightarrow s'_k = 1$$

En este caso $(s_k - s'_k) = -2$, pero para que esta actualización tenga lugar $\sum_i W_{ki} s_i > 0$, entonces $\Delta E < 0$. La otra posibilidad es:

$$s_k = 1 \rightarrow s'_k = -1$$

En este caso $(s_k - s'_k) = 2$, pero para que esta actualización tenga lugar $\sum_i W_{ki} s_i < 0$, entonces $\Delta E < 0$. Con esto finaliza la demostración.

1.2. Punto 2

Demostrar que una red de tipo Hopfield 82 que aprende un patrón también aprende su inverso.

1.2.1. Desarrollo

De antemano sabemos que si enseño a una red de Hopfield un patron como

$$\underline{\underline{W}} = \underline{P}_1 \cdot \underline{P}_1^T - \underline{\underline{I}}$$

al buscar la salida obtendré

$$\underline{P}_1 = sgn\left(\underline{\underline{W}} \cdot \underline{P}_1\right) \tag{1.2}$$

El inverso del patrón $\underline{P}_1$ es $-\underline{P}_1$ luego, debo demostrar que

$$-\underline{P}_1 = sgn\left(\underline{\underline{W}} \cdot (-\underline{P}_1)\right)$$

La función signo es una función impar, entonces, multiplicando la ecuación 1.2 por -1, obtengo:

$$-\underline{P}_1 = -sgn\left(\underline{\underline{W}} \cdot \underline{P}_1\right) = sgn\left[-\left(\underline{\underline{W}} \cdot \underline{P}_1\right)\right] = sgn\left[\underline{\underline{W}} \cdot (-\underline{P}_1)\right]$$

La matriz de pesos $\underline{\underline{W}}$ no cambió durante el proceso, por lo cual podemos decir que la red aprendió $\underline{P}_1$, pero también su inverso $-\underline{P}_1$.

Capítulo 2

Ejercicios Prácticos

2.1. Punto 1

Implemente una red de Hopfield 82 que aprenda las tres imágenes que se encuentran en la página web de la materia. Pruebe el funcionamiento de la red utilizando distintos patrones de entrada como, por ejemplo, las mismas imágenes con ruido gaussiano aditivo con distintas varianzas, y otros tipos de ruido.

2.1.1. Desarrollo

Para el punto 1 se creó un programa que lee y guarda las imágenes e imprime en pantalla las imágenes leídas y la evolución bit por bit de la imagen de salida. Durante todas las pruebas la red fue actualizada de forma asincrónica.

Lectura y Almacenamiento

Las 3 imágenes que la red aprende son torero.bmp, paloma.bmp y quijote.bmp, ver Figura 2.1.

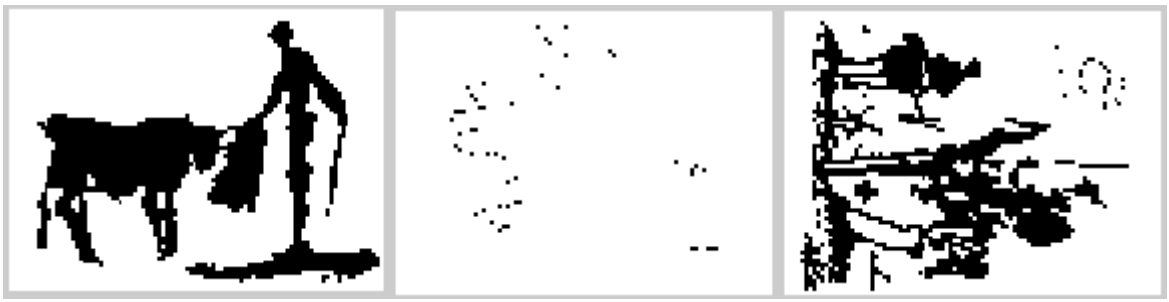

Figura 2.1: Imagenes aprende la red de Hopfield 82.

Las imágenes son leídas con la función de Matlab *imread*. Luego se modifica su tamaño en pixeles a través de la función *imresize* ya que las imágenes originales tienen una dimensión de 240 x 180 pixeles, que arregladas como vector, demandarían una matriz de pesos sinápticos de 43200 x 43200, matriz que poseería más de mil millones de datos, incapaz de ser manejada por la memoria de un computador de escritorio promedio. Todas las imágenes fueron trabajadas a una escala de 0.4 veces el tamaño original.

Aprendizaje

Para formar la matriz de pesos sinápticos se calculó:

$$\underline{\underline{W}} = \sum_{i=1}^{n} \left(\underline{p}_i \cdot \underline{p}_i^T \right) - n\underline{\underline{I}} \tag{2.1}$$

$\underline{\underline{W}}$: matriz de pesos sinápticos.
n: cantidad de patrones por almacenar.
$\underline{p}_i$: patrón.
$\underline{\underline{I}}$: matriz identidad.

Pruebas de memoria sin ruido

Se calculó la salida s_i para cada neurona de forma asincrónica mediante la regla dinámica:

$$s_i = sign\left(\sum_j W_{ij} s_j \right)$$

Se enseñaron a la red, una por una, las tres imágenes que había aprendido sin ningún tipo de ruido. Estas tres pruebas terminaron exitosamente, ya que la salida de la red, luego de un recorrido completo de todas las neuronas, arrojó un error de 0 bits frente a las imágenes originales. Esto comprueba que fueron aprendidas correctamente. En otras palabras, se le enseñaron las tres imágenes de la Figura 2.1 y la salida fueron las mismas imágenes.

Pruebas de memoria con ruido

El ruido se introdujo a todas las imágenes haciendo uso de las herramientas de dibujo del programa Paint de Microsoft. Para la primera prueba se le introdujo ruido a la imágen del torero con el pincel de aerosol, agregando puntos negros por toda la imagen. Se logró recuperar de la red la imagen original. Ver Figura 2.2.

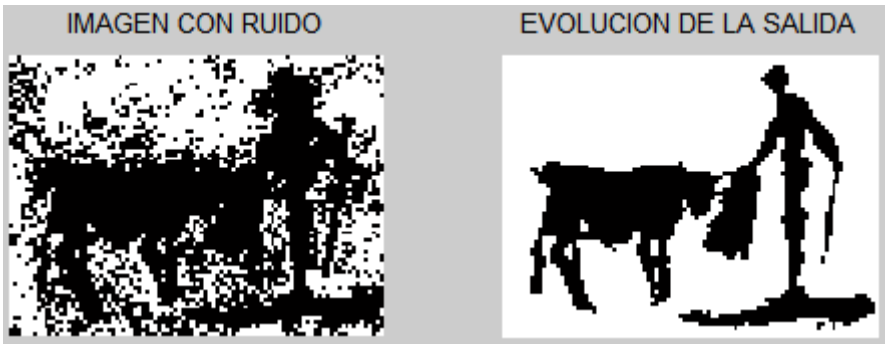

Figura 2.2: Evolución de la imagen del torero con ruido.

La imagen del torero también se probó mostrando a la red sólo la mitad superior. La imagen original pudo ser recuperada, aunque con unos pocos bits de error. Ver Figura 2.3.

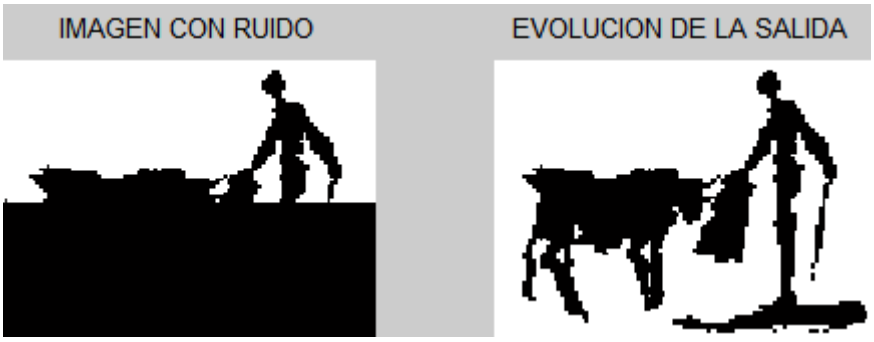

Figura 2.3: Evolución de la imagen del torero con ruido 2.

A la imagen del Quijote se le invirtieron los colores, y adicional se muestra sólo la mitad derecha sin sol y con un dibujo a la izquieda. En esta ocasión recuperó la imagen original inversa. Ver Figura 2.4.

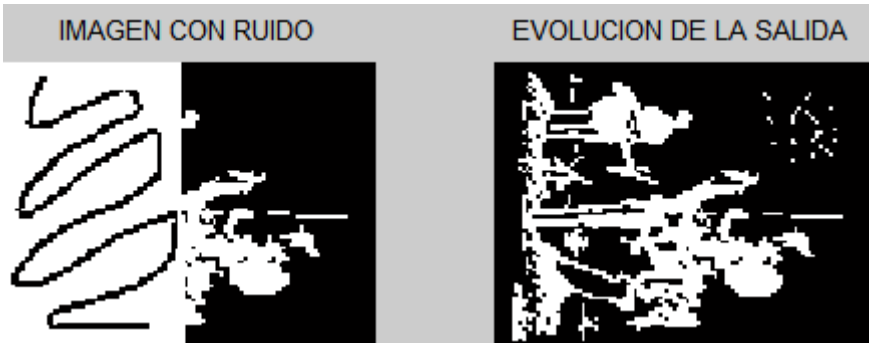

Figura 2.4: Evolución de la imagen del Quijote con ruido.

En la Figura 2.5 la imagen fue modificada removiendo la capa, la cabeza y las patas traseras del toro. La imagen fue recordada tal como aprendió la original.

A la imagen de la paloma se le invirtieron los colores. En esta ocasión se recuperó la misma imagen, demostrándose anticipadamente que las imágenes inversas también son aprendidas y son estados mínimos de energía de la red. Ver Figura 2.6.

2.2. Punto 2

Comprobar estadísticamente la capacidad de la red de Hopfield 82. Enseñar distintas cantidades de patrones pseudo-aleatorios para distintos tamaños de red y hacer una estadística de cantidad de patrones aprendidos en función del tamaño de la red.

2.2.1. Desarrollo

Se estimó la capacidad de la red de Hopfield con la pendiente de la curva Patrones aprendidos vs. Cantidad de Neuronas. Primero se generaron patrones pseudo-aleatorios, los cuales fueron enseñados a la red con la regla de aprendizaje de la ecuación 2.1. Luego, se calcularon las salidas para cada patrón y se contabilizó la cantidad de bits erróneos

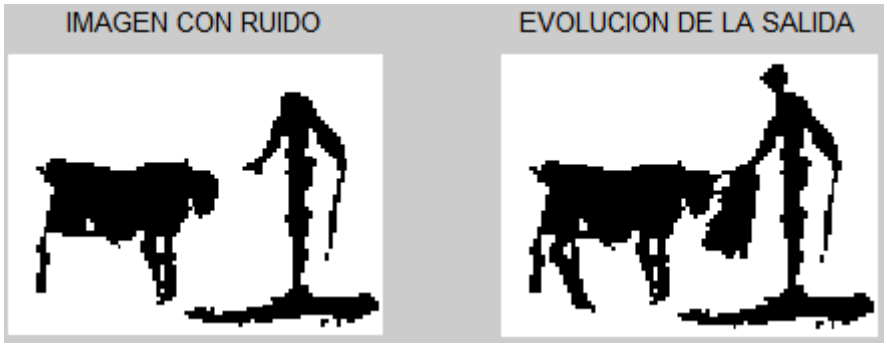

Figura 2.5: Evolución de la imagen del torero sin cabeza, capa ni patas traseras.

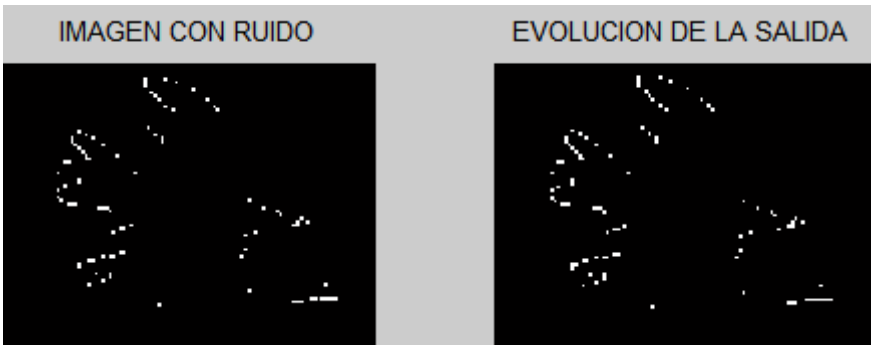

Figura 2.6: Evolución de la imagen inversa de la paloma.

acumulados para las salidas de todos los patrones. Cuando la cantidad de bits erróneos superó el 1/1000 bits de memoria total se detiene el ciclo y se registra el número de patrones aprendidos correctamente.

Este procedimiento es repetido varias veces para realizar una estadística, donde se tiene en cuenta el valor medio y la desviación estándar para cada grupo de muestras para un número de neuronas N constante. Para este informe se optó por realizar una simulación para $N = 400$ y un número de muestras M de 50. El resultado se observa en la figura 2.7, donde la capacidad, dada por una regresión lineal de los valores medios arroja $C = 0{,}0905$. Este valor es cercano al valor teórico $C = 0{,}105$, que corresponde a una probabilidad de error $P_{error} = 0{,}001$. Esta capacidad también pudo ser calculada como un promedio de capacidades, pero esta forma permite interpolar entre dos valores de N para los mismos parámetros.

Se concluye entonces que la capacidad para la red de Hopfield es constante para cualquier número de neuronas siempre que mantenga constante el error permisible.

En la figura 2.8 se observa que la capacidad de la red aumenta conforme aumenta el N. Estos datos fueron obtenidos de simulaciones donde la única variable fue el número de neuronas. Ver Cuadro 2.1.

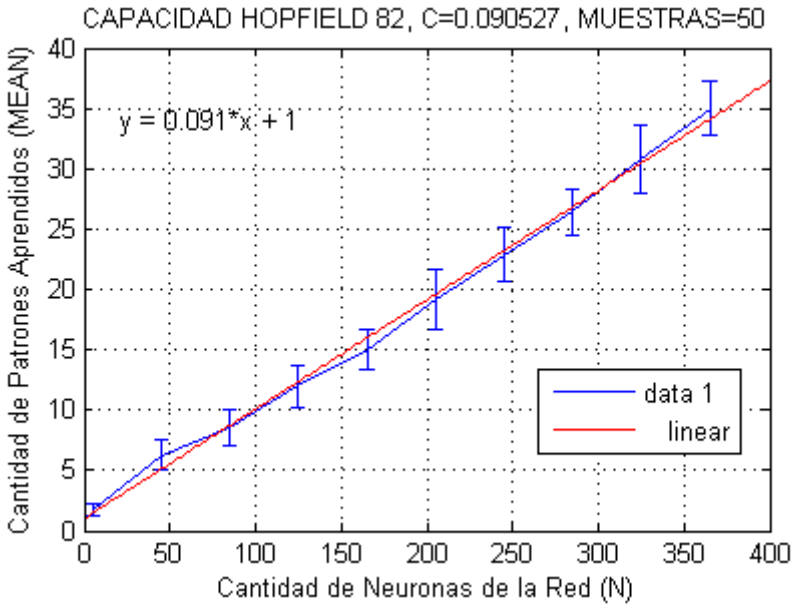

Figura 2.7: Capacidad de la red de Hopfield 82 para N=400, M=5.

Capacidad, $M = 50$	
Cantidad de neuronas N	Capacidad
100	0.0802
200	0.0837
300	0.0872
400	0.0905

Cuadro 2.1: Capacidad en función de la cantidad de neuronas N.

2.3. Punto 3

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. Deja de funcionar la red? Estime cuánto disminuye la capacidad en función de la eliminación de interconexiones.

2.3.1. Desarrollo

Procedimiento

Se enseñan los patrones a la red y luego se destruye una cantidad de conexiones fija, haciendo ceros un porcentaje de las posiciones de la matriz de pesos $\underline{\underline{W}}$. Luego se calculan las salidas estables de la red para todos los patrónes y se calcula el promedio y la desviación estndar de la cantidad de bits erróneos que se generan. A lo largo de varios ciclos *for* se va variando la cantidad de conexiones eliminadas, manteniendo constantes la cantidad de neuronas N y la cantidad de patrones aprendidos p.

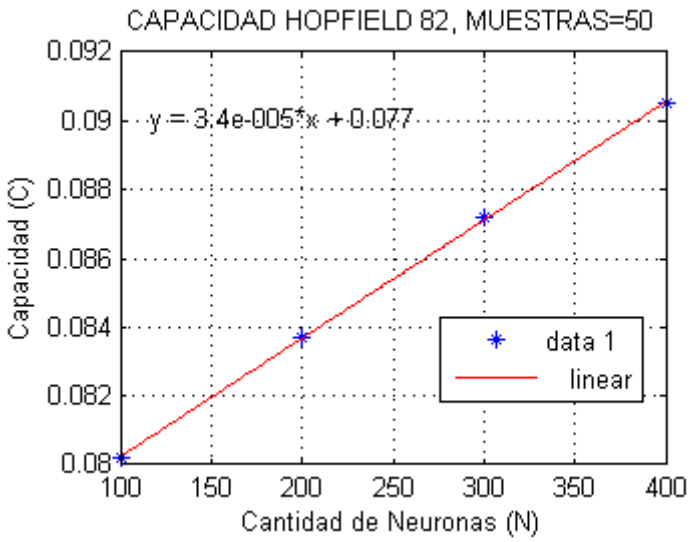

Figura 2.8: Capacidad de la red de Hopfield 82 para N variable, M=50.

Resultados

En la Figura 2.9 se aprecia el crecimiento del porcentaje de bits erróneos en función del porcentaje de conexiones eliminadas. Para este caso se tomó un número de neuronas $N = 200$ y una cantidad de patrones apredidos de $p = 10$, de tal manera que no fuera superada la capacidad para el error permisible que se trabajó, que fue de un 1 bit erróneo por cada 1000 bits de memoria total. Esta capacidad es $p_{max}/N = 0{,}105$ para $p_{error} = 0{,}001$, de modo que eligiendo así N y p garantizamos que la red aprende el 100 por ciento de los patrones, ya que $p/N = 10/200 = 0{,}05$.

Se pude concluir que la capacidad de la red disminuye conforme aumenta la cantidad de conexiones eliminadas de la matriz de pesos W. La cantidad de errores de bits diferentes al comparar con el patrón aprendido se mantiene baja hasta alrededor de un 60 por ciento de conexiones eliminadas y aumenta exponencialmente a medida que se acerca al 100 por ciento, lo que indica que tiene muy buena capacidad siempre que los patrones aprendidos estén por debajo del limite de capacidad y las conexiones eliminadas no superen el 60 por ciento aproximadamente.

2.4. Punto 4

Verifique la existencia de estados espurios producidos por patrones inversos y mezclas (Ver *Spurious States*, Hertz, Krogh y Palmer, pág. 24).

2.4.1. Desarrollo

Estados espurios producidos por patrones inversos

El algoritmo de Matlab usado para el punto 2 es útil para comprobar que existen dichos estados. A una red de tamaño variable se le enseñan patrones y luego se calcula el inverso de todos los patrones. Posteriormente se comprueba la capacidad patrón por

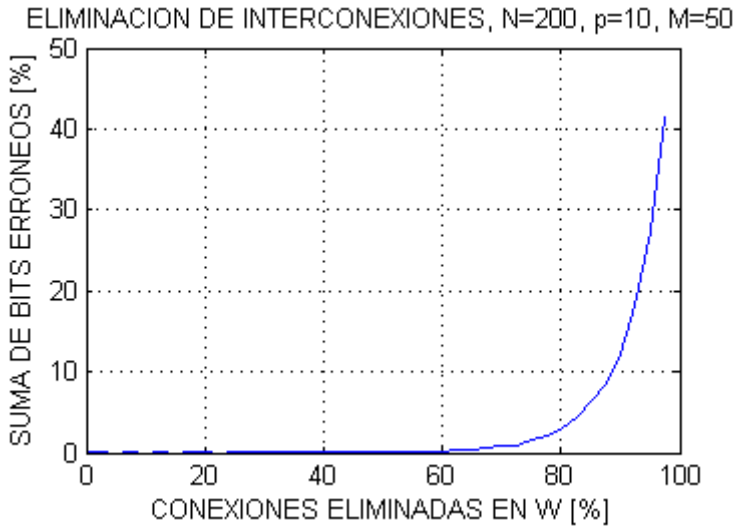

Figura 2.9: Capacidad de la red de Hopfield en función de la variación de interconexiones eliminadas de $\underline{\underline{W}}$.

patrón con los patrones inversos, incrementando número de patrones y tamaño de la red N. Los resultados obtenidos son muy similares en comparación con el cálculo de la capacidad utilizando los patrones no inversos (es decir, los originales). Esto se ilustra con la Figura 2.10.

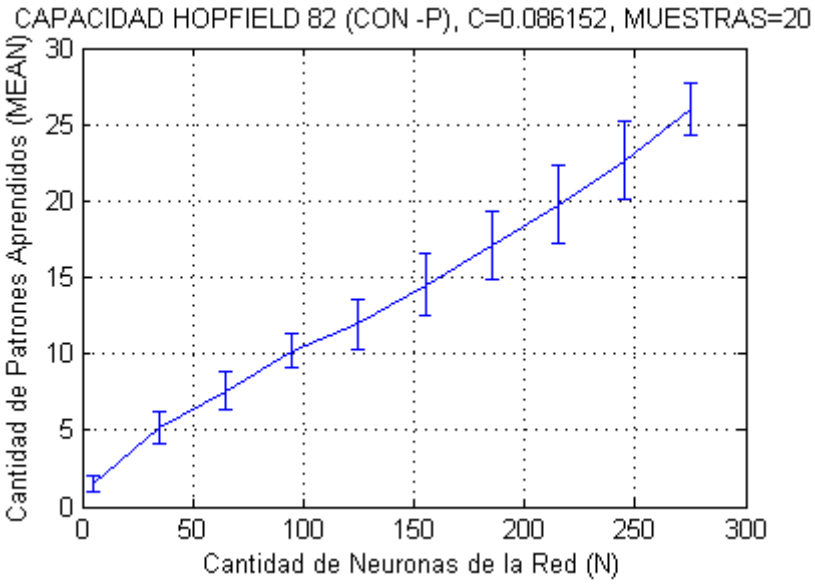

Figura 2.10: Capacidad calculada enseñando solo los patrones inversos.

Estados espurios producidos por mezclas de patrones

El algoritmo diseñado genera NP patrones aleatorios, los enseña a la red, calcula $NCOMB$ mezclas y luego las muestra a la red una por una para calcular la cantidad de bits erróneos totales para todos los patrones. Se optó por graficar en el eye Y la suma de

bits erróneos sobre la cantidad total de bits, que resulta de multiplicar $NP \times N$. Esto con el fin de facilitar la visualización al tener la cantidad de bits erróneos como una fracción de la cantidad de bits totales.

Las mezclas fueron generadas a partir de un grupo original de patrones, del cual se van extrayendo subgrupos de 3, 5 y 7 patrones. Cada subgrupo es mezclado (sumado) y operado por la función signo para así generar un nuevo patrón. Más tarde estos nuevos patrónes son mostrados a la red para obtener la cantidad de bits erróneos que arroja la memoria.

En la Figura 2.11 se puede observar el porcentaje de bits erróneos en función de la cantidad de neuronas N. Para realizar la gráfica se enseñaron a la red 40 patrones, se generaron 20 mezclas y se generaron 20 muestras con todos los parámetros constantes para la estadística. Todos los estados graficados lograron un mínimo de energía, lo que los ubica en un mínimo local. Esto demuestra que las mezclas impares de patrones son estados espurios y que la red tiende a reconocerlos con cero error mientras mayor sea su tamaño. Esto también es evidencia de que la capacidad de la red también aumenta para las mezclas de patrones conforme aumenta N.

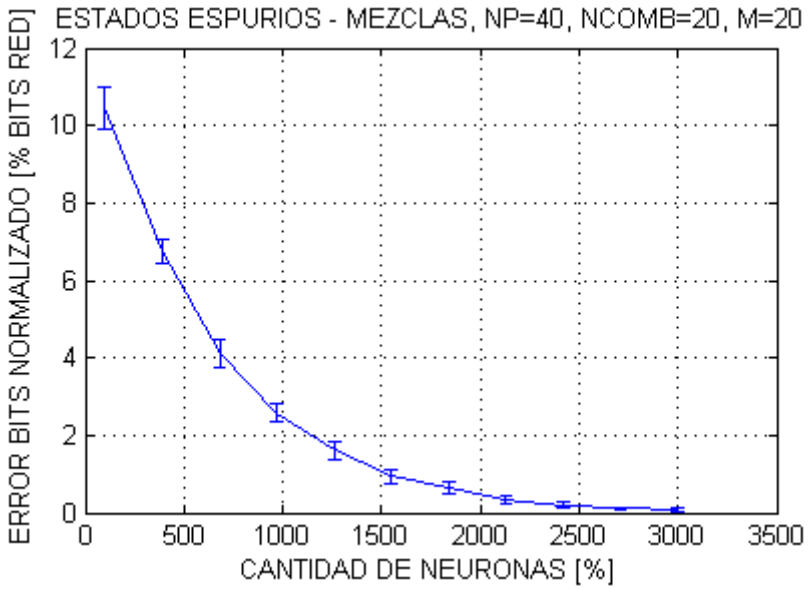

Figura 2.11: Bits erroneos al mostrar mezclas aleatorias de 3, 5 y 7 patrones a la red.

2.5. Punto 5

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

2.5.1. Desarrollo

Para el desarrollo de este punto se empleó el algoritmo de Metrópolis para estudiar el modelo de evolución ferromagnética de Ising mediante una simulación numérica. El

desarrollo de este punto del trabajo práctico siguió un punto de vista distinto para generar la curva de magnetización, donde para cada temperatura se corre el algoritmo de Metrópolis partiendo siempre de un estado de espines desordenado. Esta simulación es equivalente a ir descendiendo la temperatura y tomar el estado de temperatura anterior como el estado actual de la red. Ver la referencia [1].

El modelo de Ising permite estimar una temperatura de Courie (llamada también temperatura crítica) a partir de la cual un material magnético pierde su magnetización cuando la misma pasa a estar dominada por la temperatura. Si fijamos una temperatura menor que T_c la magnetización se estabiliza en valores cercanos a 1, y en el caso contrario la magnetización tiende a ser cero.

El algoritmo de Metrópolis consta de los siguientes pasos:

1. Escoger un espín σ_k al azar de la retícula.
2. Calcular el delta de energía como:

$$\Delta E = 2J\sigma_k^0 \sum_{i \ cercanos} \sigma_i^0$$

   J: es la interacción entre espines. Se asume constante.
   σ_k^0: valor de espin en la posición k antes del cambio de spin.
   σ_i^0: espin vecino de σ_k^0.
3. Si $\Delta E \leq 0$ se acepta inmediatamente el cambio $\sigma_k \rightarrow -\sigma_k$.
4. Si $\Delta E > 0$ se genera un número aleatorio uniforme P_a en el intervalo $[0, 1)$.
5. Si $P_a < e^{-\Delta E/kT}$ se da igualmente el cambio $\sigma_k \rightarrow -\sigma_k$.
   k: constante. Para este modelo se asumió igual a 1.
   T: temperatura. Es variable para el problema.
6. En cualquier otro caso el sistema no se altera.
7. Se repite para otros espines σ_k de la retícula.

Modelo de Ising en 1D

Para correr este modelo en Matlab se especificó la cantidad de espines, la cantidad de intercambios completos de la red de espines (se utilizó la función *randperm()* de Matlab) y la cantidad de pasos de temperatura. Los resultados están ilustrados en la Figura 2.12. All se aprecia que no existe una temperatura T_c de Courie que podamos identificar como temperatura a partir de la cual el modelo se desmagnetiza, a pesar de que se buscó un estado de equilibrio con un número de pasos por red (recorridos completos por todos los espines) de 10000. La cantidad de espines de la malla unidimensional es de 100. Como conclusión podemos decir que para el modelo unidimensional no se alcanza magnetización para ninguna temperatura.

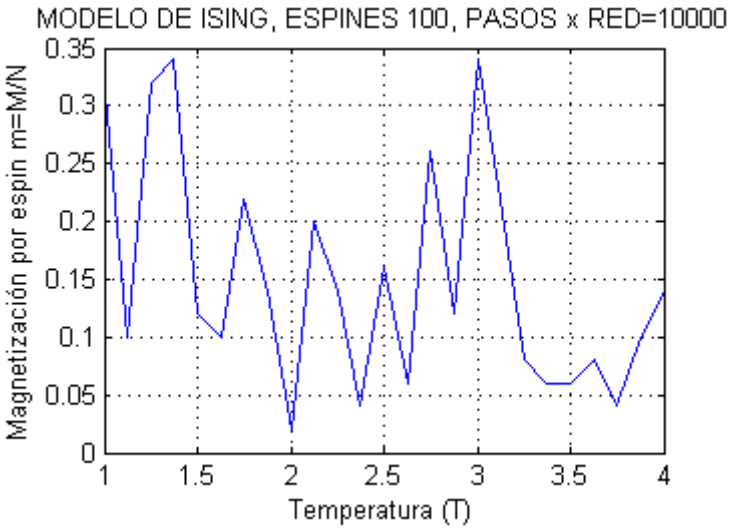

Figura 2.12: Magnetización ferromagnética por espin para una malla 1D de 100 elementos. $i = 10000$.

Modelo de Ising en 2D

Para correr este modelo en Matlab se especifica la cantidad de espines por lado de la retícula y la cantidad de barridos aleatorios totales de la red de espines. Para esto se emplea la función *randperm()* de Matlab.

Se elaboró una simulación para ilustrar el proceso de magnetización, iniciando con una distribución aleatoria de espines que evoluciona hasta magnetizarse por completo para una temperatura $T = 1$. El inicio del proceso se muestra en la Figura 2.13. Aquí los espines están totalmente desordenados y no se ha iniciado el algoritmo de Metrópolis.

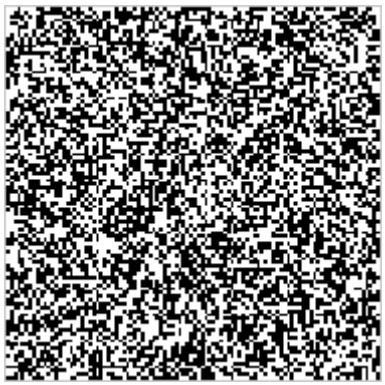

Figura 2.13: Estado inicial de magnetización ferromagnética para una mapa de espines de 100 x 100. $i = 0$, $T = 1$.

Posteriormente, el algoritmo es iniciado, obteniéndose una magnetización parcial para una cantidad de barridos completos de la red $i = 20$. Ver Figura 2.14. Para un $i = 50$ la red ha evolucionado hasta un estado de magnetización mayor, ver Figura 2.15. Finalmente, en la Figura 2.16 se observa el progreso de la magnetización con $i = 200$, donde

se observa una clara tendencia a magnetizarse por completo. Para este modelo con $T = 1$ se logró una magnetización por espin $m = M/N$ muy cercana a 1, donde M es la suma de todos los estados y N la cantidad de espines. Para este tamaño de red $i = 1000$ fue suficiente para lograrlo.

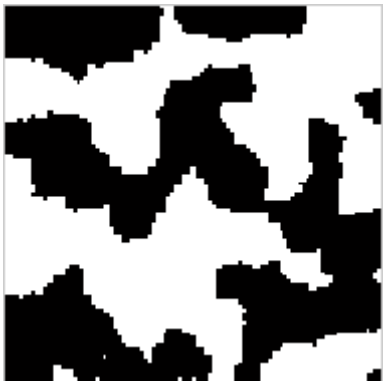

Figura 2.14: Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 20$, $T = 1$.

Figura 2.15: Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 50$, $T = 1$.

Por último, en la Figura 2.17 se grafican los resultados de la magnetización por espin en función de la temperatura. El tamaño de la malla de espines es de 30 x 30 y se corrieron 1000 recorridos de intercambio por toda la red de espines para cada temperatura (24 en total), la cual fue variada entre 1 y 4. En la figura se puede ver como la magnetización se mantiene cercana a 1 para temperaturas por debajo de la temperatura de Courie T_c, que tiene un valor teórico de 2.269185, y cercana a cero para valor que están por encima. Esto indica que el modelo de magnetización no puede sostenerse magnetizado a estas temperaturas y los espines nunca logran alinearse en una sola dirección (es decir, que tengan casi todos el mismo signo).

Figura 2.16: Magnetización ferromagnética para una mapa de spins de 100 x 100. Barridos $i = 200$, $T = 1$.

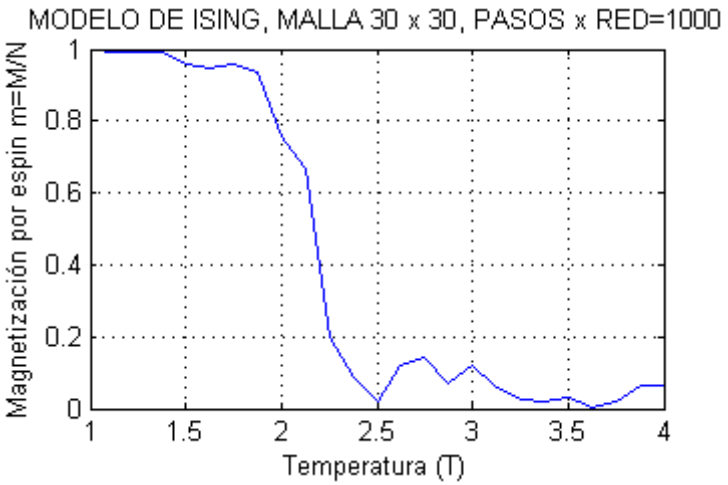

Figura 2.17: Magnetización ferromagnética por espin para una mapa de spins de 30 x 30. Barridos $i = 1000$.

Capítulo 3

Comentarios y Conclusiones

- Los problemas de memoria de los computadores de escritorio empleados para el desarrollo de los ejercicios no permitieron realizar los ejercicios del punto 1 con las imágenes originales de 240 x 180 pixeles. Una forma de tratar este problema es emplear un formato distinto para el almacenamiento y las operaciones de las matrices y los vectores, como es, por ejemplo, el formato .mtx, que permitiría darle un manejo más eficiente a la memoria.

- Para el punto 1 se implementó una red de Hopfield 82 que aprende tres patrones. Al mostrarle los patrones enseñados con algo de ruido logra sacar como salida el mismo patrón sin ruido.

- Para el punto 2 se evaluó la capacidad de la red. Con $N = 400$ neuronas se logró una capacidad de $C = 0{,}0905$, cercana al valor teórico de 0,105. Es claro que este valor teórico se alcanza para un número muy grande de neuronas, mucho mayor que $N = 400$.

- El modelo de Ising 2D, cuyos resultados se obtuvieron iterando con el algoritmo de Metrópolis, arroja que para temperaturas por debajo de la temperatura de Courie T_c la malla de espines tiende a magnetizarse $m = 1$, mientras que para temperaturas por encima de T_c la malla tiende a permanecer desmagnetizada $m = 0$. La solución 2D puede compararse con la solución de Onsager. No se pudo obtener una curva de magnetización para el caso 1D que señalara una temperatura crítica. Esto para cualquier temperatura y cualquier cantidad de intercambios de espines.

- Las mezclas impares de patrones también son aprendidas por la red de Hopfield 82. Esto se comprobó numéricamente al lograr estados de mínima energía al mostrar a la red patrones que fueron el resultado de sumas impares de los patrones enseñados a la red.

Bibliografía

[1] S. M. Del Razo and Alejandro Guayaquil Sosa. Simulación del modelo de ising en una retícula 2d cuadrada con condiciones perioódicas. *Facultad de Ciencias, UNAM.*