



SISTEMAS DE INTELIGENCIA ARTIFICIAL

TRABAJO PRÁCTICO N°3

Redes de Hopfield

Autores:

Pablo Ballesty - 49359

Nicolás Magni - 48008

Guillermo Liss - 49282

22 de mayo de 2012

Resumen

El objetivo del presente informe es detallar las decisiones tomadas durante el diseño e implementación de redes de Hopfield, para ser utilizadas como memorias direccionables por el contenido. Como patrones se utilizan 25 imágenes binarias provistas por la cátedra. Se presentan resultados y conclusiones obtenidas.

1. Desarrollo

En las siguientes secciones se detallan los aspectos que se consideraron destacables durante el desarrollo del trabajo.

1.1. Evaluación y condición de corte

Siguiendo el modelo de Hopfield la evaluación se realiza de forma asincrónica. Se asegura que luego de 4096 rondas, todas las neuronas han actualizado su estado. De esta forma, se considera que el patrón de entrada convergió a un estado dado, cuando las 4096 rondas arrojan el mismo resultado, llegado este punto se corta la evaluación.

1.2. Estabilidad de patrones

Para probar la estabilidad de los patrones almacenados en una red, se implementó un *script* que calcula el *crosstalk* para cada posición del patrón. A continuación, se describe la regla para considerar a un patrón estable o inestable.

$$\text{estable}(\xi^\mu) = \begin{cases} \text{verdadero} & \forall i \text{ crosstalk}(\xi_i^\mu) < 1 \\ \text{falso} & \text{en otro caso.} \end{cases} \quad (1)$$

Luego, se utiliza esta regla para definir si un conjunto de patrones resulta estable o inestable. A continuación se presenta la misma.

$$\text{conjuntoEstable}(\xi^{\mu_n}) = \begin{cases} \text{verdadero} & \forall n \text{ estable}(\xi^{\mu_n}) == \text{verdadero} \\ \text{falso} & \text{en otro caso.} \end{cases} \quad (2)$$

Se entiende a ξ^{μ_n} como un conjunto finito de patrones $[\xi^{\mu_1} \xi^{\mu_2} \dots \xi^{\mu_n}]$ a almacenar.

De esta forma, si el conjunto de patrones almacenados resulta estable, se asegura que estos patrones son verdaderos atractores de la red.

1.3. Estados espurios

Se desarrollaron dos *scripts* diferentes, con el fin de generar patrones que puedan caer en cuencas de atracción de estados espurios de la red. A continuación se describen los mismos:

- **invert:** invierte el patrón que se le indica. Se utiliza con los patrones que se han almacenado en la red, a fin de observar si los patrones invertidos convergen a estados espurios de la red.
- **mixPatterns:** genera todas las posibles combinaciones lineales con coeficientes en 1 de los patrones que se le indican. A continuación se presenta la ecuación utilizada para definir el patrón mezcla, donde \pm significa que se realizan todas las posibles combinaciones de $+$ y $-$.

$$\forall i \quad \xi_i^{\text{mix}} = \text{sign}(\pm \xi_i^{\mu_1} \pm \xi_i^{\mu_2} \dots \pm \xi_i^{\mu_n}) \quad (3)$$

1.4. Remoción de estados espurios

Se implementó un *script* para intentar remover estados espurios que se encuentran al momento de evaluar la red con diferentes patrones. El mismo, consiste en realizar los siguientes pasos:

1. Se aplica a todos los pesos, el término corrector de la ecuación 4 siendo S^f el estado espurio que se desea eliminar.
2. Se evalúa el patrón que convergía a ese estado espurio, si este sigue convergiendo se vuelve al paso 1.

$$\Delta w_{ij} = -\frac{\epsilon}{N} S_i^f S_j^f \quad (4)$$

1.5. Cuencas de atracción

Para analizar las cuencas de atracción de los patrones almacenados y los estados espurios en la red, se implementó el *script* **noise** el cual aplica ruido sobre el patrón indicado con una densidad d . Este mismo, se utiliza en combinación con **invert** y **mixPatterns**, a fin de cubrir gran parte del dominio de entrada, y poder analizar la convergencia en cada caso.

1.6. Cantidad máxima de patrones almacenados

Si se considera a un patrón como un conjunto de variables aleatorias independientes, donde la variable aleatoria puede tomar el valor 1 o -1 con igual probabilidad. Se define P_{error} como la probabilidad de que un *bit* elegido sea inestable. Esta puede estimarse, de la siguiente forma

$$P_{\text{error}} = \text{Prob}(C_i^\nu > 1). \quad (5)$$

donde

$$C_i^\nu = -\xi_i^\nu \frac{1}{N} \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu. \quad (6)$$

El término especificado en la ecuación 6, depende de p (cantidad de patrones) y N (cantidad de neuronas). Si se consideran N y p grandes, por la ley de los grandes números, puede aproximarse la distribución binomial de C_i^ν como una distribución normal. Luego,

$$P_{\text{error}} = \text{Prob}(C_i^\nu > 1) = \frac{1}{2} [1 - \text{erf}(\sqrt{N/2p})] \quad (7)$$

donde

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-u^2) \delta u.$$

Luego, considerando aceptable un $P_{\text{error}} = 0.01$, se obtiene la relación $p_{\text{max}} = 0.15N$. Por lo tanto, en nuestro caso como $N = 4096$, la cantidad máxima almacenable por la red es

$$p_{\text{max}} \simeq 614. \quad (8)$$

Por lo tanto, teóricamente la red puede almacenar 614 patrones. Sin embargo, los patrones que se poseen, no son un conjunto de variables independientes, por lo que se implementó el *script* **bestCombination** el cual busca el conjunto más grande de patrones estables, con el fin de comparar la cantidad teórica y la cantidad real de patrones que se pueden almacenar.

2. Resultados

2.1. Prueba 1

Para todas las pruebas en esta sección se utiliza una red que almacena los siguientes patrones

- a.png
- line2.png
- windows.png

los mismos pueden verse en la figura 1.

2.1.1. Prueba de estabilidad

Utilizando la regla *conjuntoEstable* se obtiene que el conjunto de patrones resulta estable. Se evalúa la red utilizando como entrada, los patrones almacenados. Se muestran los resultados obtenidos en la figura 2.

2.1.2. Prueba de atracción

Para la misma red del caso anterior, se prueban como entrada los patrones almacenados pero con diferentes densidades de ruido.

- En la figura 3 se presentan las entradas y los resultados obtenidos, habiendo aplicado un ruido con una densidad del 10 %.
- En la figura 4 se presentan las entradas y los resultados obtenidos, habiendo aplicado un ruido con una densidad del 30 %.
- En la figura 5 se presentan las entradas y los resultados obtenidos, habiendo aplicado un ruido con una densidad del 50 %.

También se prueban como entrada versiones incompletas de los patrones almacenados en la red. Se presentan los resultados obtenidos en las figuras 6 y 7.

Además, se prueban patrones que no fueron almacenados en la red. Se tomaron

- h.png
- line1.png
- mac.png

. Se muestran los resultados en la figura 8.

2.1.3. Prueba de patrones inversos

Se prueban las salidas que arroja la red, para las entradas conformadas por los inversos de los patrones almacenados en la red. Se muestran los resultados en la figura 9.

2.1.4. Prueba de patrones mezcla

Se prueban las salidas que arroja la red para entradas conformadas por mezclas de los patrones almacenados en la red. Se muestran las entradas y los resultados en la figura 10.

2.1.5. Remoción de estados espurios

Se intentan remover los estados espurios a los que llega la red, se pueden ver los resultados obtenidos utilizando entradas inversas y mezcla de los patrones almacenados. Luego de realizar los cambios en la red, se evalúan las entradas que convergían a esos estados, como también los patrones originales.

En la figura 11 se muestran los resultados obtenidos, luego de remover el estado conformado por el inverso del patrón *a.png*.

En la figura 12 se muestran los resultados obtenidos, luego de remover el estado conformado por la mezcla de los patrones almacenados en la red.

2.1.6. Prueba de energía

En la figura 13, se muestra el gráfico de la energía correspondiente a la evaluación de la entrada *a.png* con ruido del 50 %.

2.2. Prueba 2

Se utilizó el *script bestCombination* para hallar el conjunto más grande de patrones estables. Se encontró un conjunto conformado por los siguientes siete patrones

- a.png
- bad-egg.png
- circle-union.png
- circle1.png
- footprint.png
- line1.png
- midnight-bsd.png.

En la figura 14 se muestran los mismos.

3. Conclusiones

3.1. Memoria direccionable por el contenido

Se pudo observar que la red que almacena un conjunto de patrones estables funciona como una memoria direccionable por el contenido. En las figuras 2, 3, 4 y 5 puede notarse que los patrones almacenados funcionan como verdaderos atractores, y estos poseen cuencas de atracción considerables.

3.2. Patrones atractores

Pudo notarse que además de existir los atractores correspondientes a los patrones almacenados en la red, existen otros atractores correspondientes a estados espurios. Puede verse en las figuras 8, 9 y 10 que diferentes entradas convergen a estados espurios de la red.

Se pudo notar, que los estados espurios correspondientes a los patrones almacenados invertidos poseen cuencas de atracción muy grandes y comparables con las de los patrones originales. Mientras que, los estados espurios conformados por la mezcla de patrones almacenados poseen una cuenca de atracción pequeña.

3.3. Eliminación de estados espurios

Se pudo notar que al eliminar estados espurios de la red se puede afectar también a los patrones almacenados. Puede verse en la figura 11 que al tratar de eliminar un estado espurio correspondiente al inverso de un patrón almacenado, luego si se utiliza el almacenado como entrada, ya no converge al atractor correspondiente.

También se pudo observar en la figura 12, que al eliminar un estado espurio correspondiente a una mezcla de los patrones almacenados, también se elimina el estado espurio inverso.

3.4. Decrecimiento de la energía

Como era esperable, en la figura 13 se puede observar que en la evaluación la energía siempre decrece, que es lo correspondiente en un modelo de Hopfield, ya que las actualizaciones se realizan de forma asincrónica.

3.5. Capacidad de almacenamiento

Como se calculó, teóricamente la red puede almacenar como máximo 614 patrones distintos. Sin embargo, este cálculo está basado en que los patrones son un conjunto de variables aleatorias independientes. Se pudo notar que en la práctica la cantidad de patrones a almacenar depende de la similitud con respecto a la distancia de Hamming que poseen los patrones que se desean almacenar, y que la cantidad máxima es mucho menor a la calculada de forma teórica. En este caso, la cantidad máxima de patrones que se pudo almacenar fue de 7 patrones.

A. Imágenes

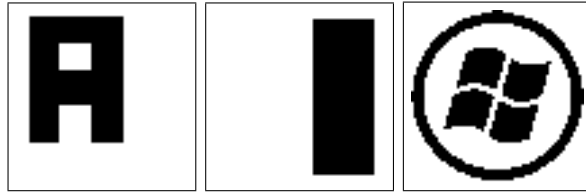


Figura 1: Patrones almacenados en la prueba de estabilidad

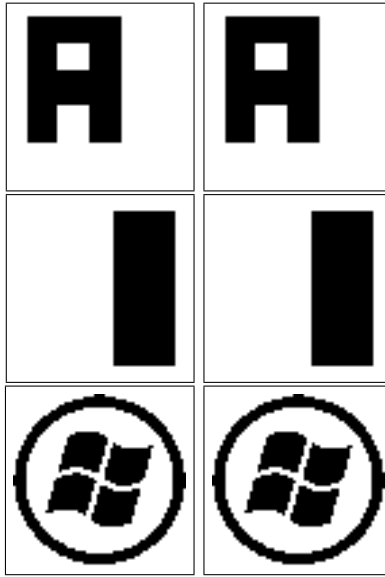


Figura 2: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

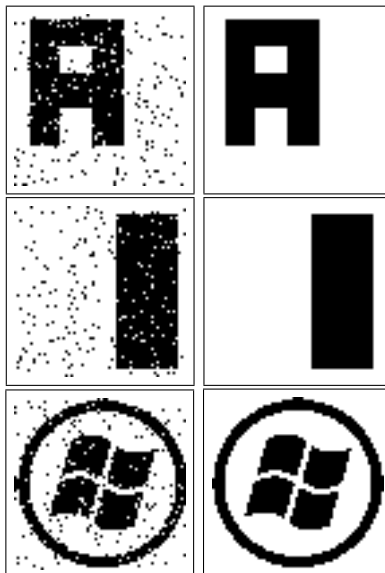


Figura 3: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

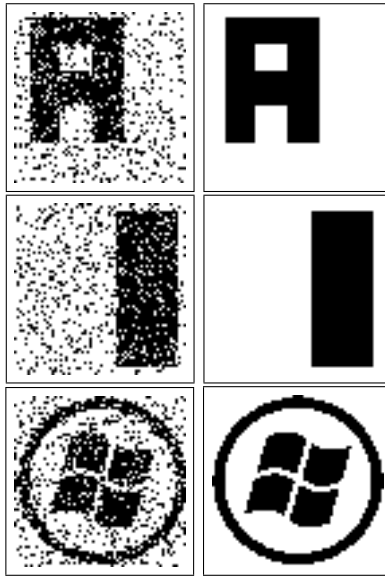


Figura 4: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

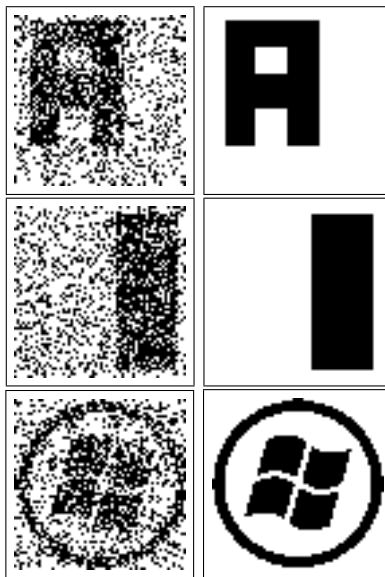


Figura 5: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

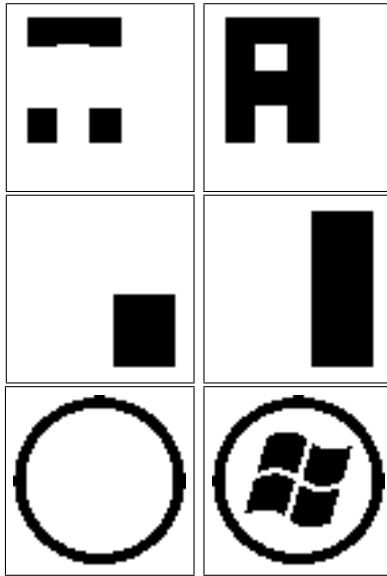


Figura 6: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

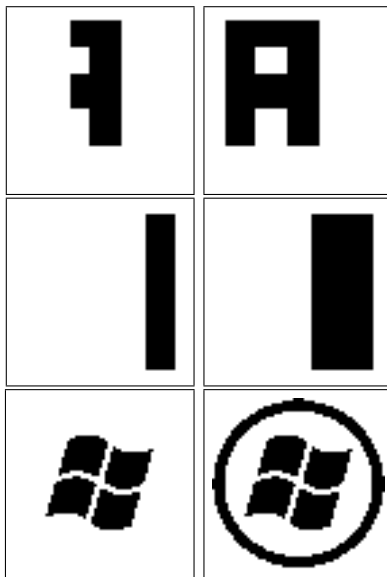


Figura 7: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

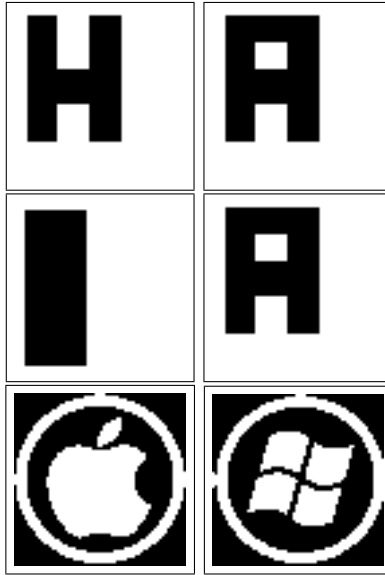


Figura 8: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

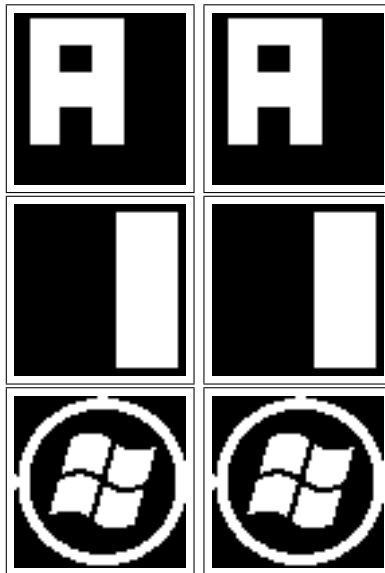


Figura 9: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.



Figura 10: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

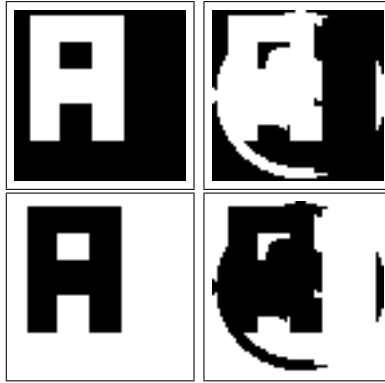


Figura 11: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

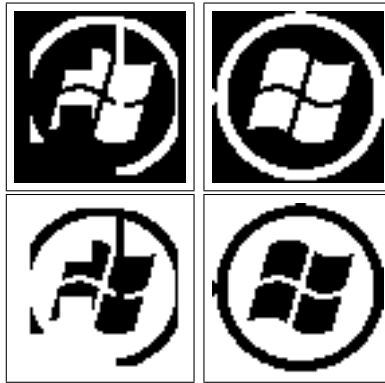


Figura 12: A la izquierda se presenta la entrada de la red, y a la derecha la salida obtenida.

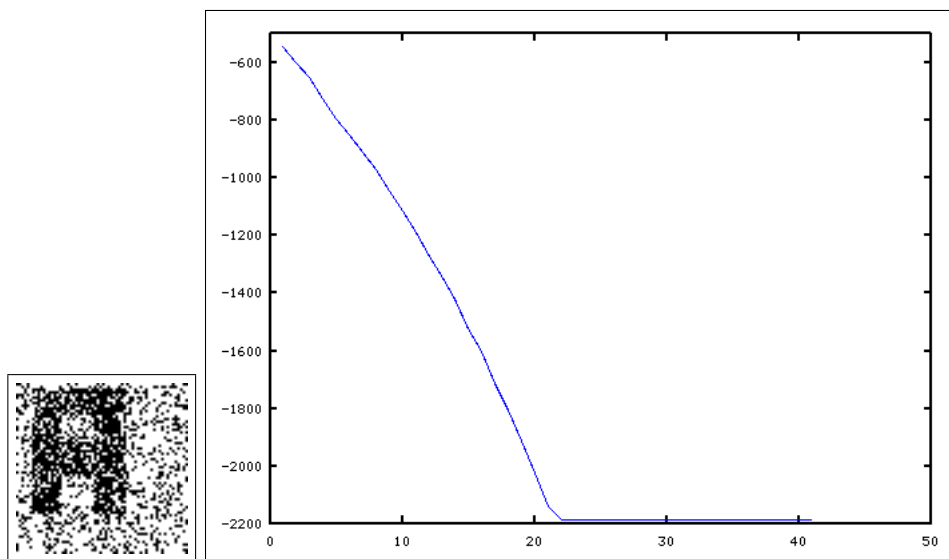


Figura 13: Energía en la evaluación de la entrada presentada a la izquierda.

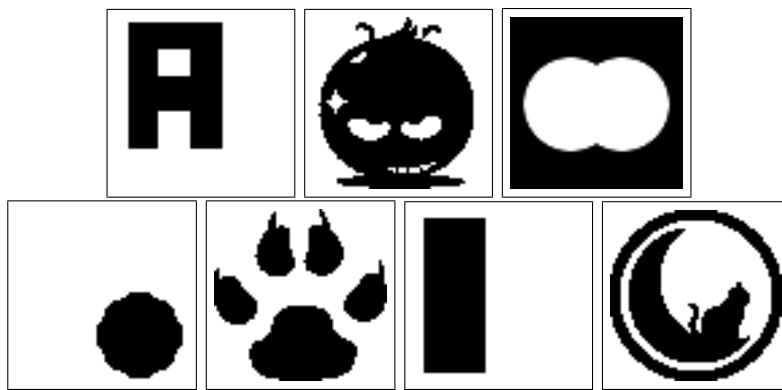


Figura 14: Conjunto de patrones estables más grande.