Redes neuronales multicapa

Castiglione, Karpovsky, Sturla

Sistemas de Inteligencia Artificial

3 de Mayo de 2012

Tabla de contenidos

El problema

El problema consistió en analizar el comportamiento de una Red de hopfield como memoria asociativa direccionable por el contenido.

- Red de Hopfield implementada en el lenguaje *Java*.

Modelado del problema

Algunas definiciones preliminares:

- Ψ: Conjunto de patrones a memorizar
- N: Longitud de los patrones de entrada (ej. $N = 64$ si cada imagen es de $8x8$ pixels)
- p: Cantidad de patrones distintos que contiene Ψ

Se representó a la Red de Hopfield como una clase abstracta debido a la distinción entre el algoritmo **sincrónico** y el algoritmo **asincrónico**.

Patrones

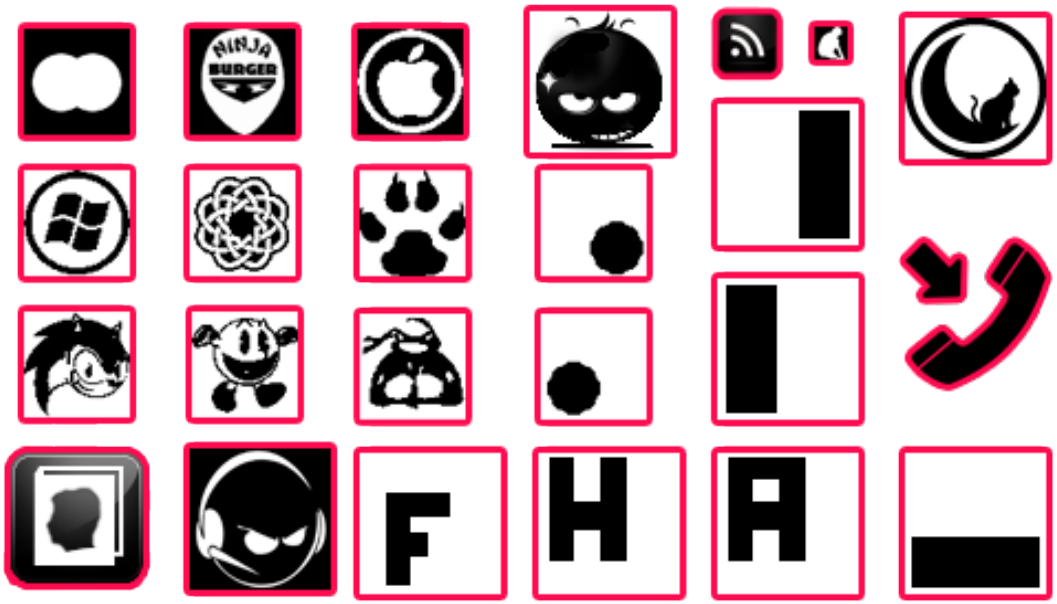

Figura 1: Visualización de los 25 patrones con la que se inicializa a la red

Modelado del problema

La clase contiene dos variables: un vector y una matriz que están definidos como sigue:

- **float[N][N] weights**: Matriz de pesos w_{ij} correspondientes a los patrones que se desean memorizar.
- **int[N] states**: Vector con los estados de cada una de las N neuronas.

Nótese que si bien el vector de estados es de tipo entero, los estados definidos para este problema son 1 o -1 dado que las unidades que están siendo utilizadas son **bipolares**.

Modelado del problema

- Representación de μ

Representar a los patrones μ como vectores de enteros **int[] pattern** los cuales contienen un 1 en la posición *i* en el caso de que el i-ésimo pixel del patrón sea negro o contienen un -1 en la posición *i* en el caso de que el i-ésimo pixel del patrón sea blanco.

- Los patrones fueron *normalizados* (escala de grises a B&W)

Tabla de contenidos

Inicialización de la red

El método *storePatterns* inicializa la matriz de pesos sinápticos de la red utilizando la regla del producto externo de **Hebb**.
Nótese que la matriz de pesos, una vez inicializada, no cambia durante toda la ejecución del algoritmo.

- **Simetría:** La matriz de pesos es simétrica con $w_{ij} = 0$
- **Matriz de pesos:** Una vez inicializada queda estática.

Función de activación

La función de activación utizada es la función signo: $\mathbf{f(x) = x}$.

Actualización de estados

Regla de actualización

$$S_i(n+1) = sgn\left(\sum_{j=1}^{N} W_{ij} S_j(n)\right)$$

La convergencia está dada cuando el vector de estados permanece invariante respecto del vector en el paso anterior o se detecte un **ciclo de longitud dos** en las actualizaciones del vector de estados.

Red sincrónica: Modelo de Little

Se implementó una red neuronal con el modelo de **Little** en la cual la actualización de los estados es sincrónica.

Actualización de estados

- Actualizar todas las neuronas simultaneamente en cada paso.

Red asincrónica: Modelo de Hopfield

Se implementó una red neuronal con el modelo de **Hopfield** en la cual la actualización de los estados es asincrónica.

Actualización de estados

- En cada paso se actualiza de a una unidad por vez, la misma es elegida al azar.

Mejoras implementadas

- Bit de paridad
- Matriz pseudo-inversa

Matriz pseudo-inversa

El cálculo de la matriz pseudo-inversa viene dado por:

Cálculo de matriz pseudo-inversa

$$w_{ij} = \frac{1}{N} \sum_{\mu\nu} \xi_i^\mu (Q^{-1})_{\mu\nu} \xi_j^\nu$$

siendo

$$Q_{\mu\nu} = \frac{1}{N} \sum_i \xi_i^\mu \xi_i^\nu$$

Matriz pseudo-inversa

El error de la imagen que se obtiene como respuesta está dado por:

Cálculo del error

$$error = \frac{bits\_incorrectos}{bits\_totales}$$

Tabla de contenidos

Estados espúreos

Se utilizó una red asincrónica y se la hizo memorizar 14 imágenes distintas. El resultado de pasarle como entrada la imagen “line1” se puede observar a continuación:

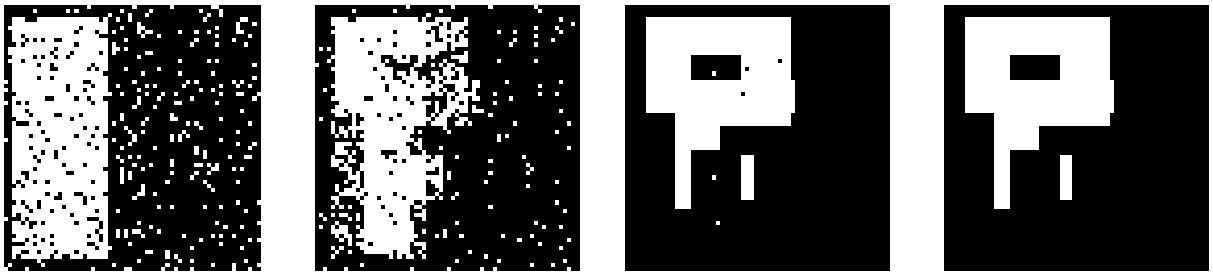

Figura 1: Ejemplo de estado espúreo.

Patrones con ruido o incompletos

Se utilizó una red asincrónica y se la hizo memorizar los archivos “foot”, “line1”, “line2”, “line3”, “line4”.
A continuación se muestra el resultado de pasarle como entrada la imagen “foot” pero incompleta.

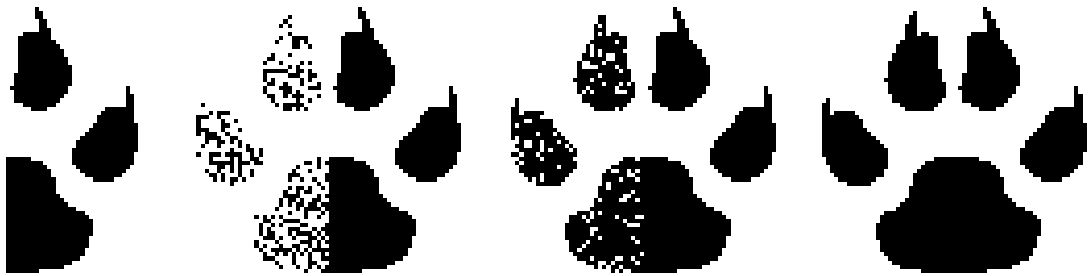

Figura 1: Respuesta de una red de hopfield asincrónica ante un patrón incompleto.

Patrones con ruido o incompletos

Si en cambio se utiliza solamente ruido en una red asincrónica que memorizó los patrones “f”, “h”, “a”, “footprint” y le pedimos que evolucione partiendo de “f”, la red rápidamente converge a la imagen deseada:

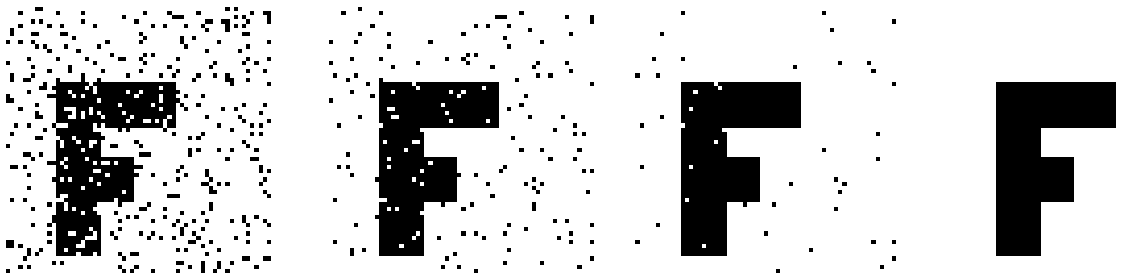

Figura 1: Respuesta de una red de hopfield asincrónica ante un patrón con ruido.

Patrones invertidos

Red inicializada con las siguientes imágenes provistas por la cátedra: “line1”, “line2”, “line3”, “line4”, “h”.

- Mejora del bit de paridad
- Mejora de pseudo-inversa
- Patrón de entrada invertido y con ruido

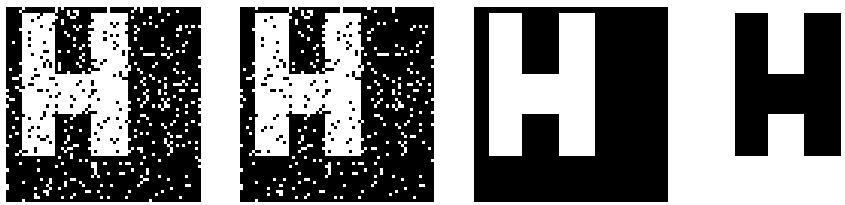

Figura 1: Estados intermedios para una red de hopfield asincrónica con la mejora del bit de paridad y la utilización de la matriz pseudo-inversa

Patrones desconocidos

El siguiente ejemplo muestra la respuesta de la red ante un patrón desconocido. Se utilizó una red asincrónica y se la hizo memorizar los archivos “line1”, “line2”, “line3”, “line4”

.

A continuación se muestra el resultado de pasarle como entrada la imagen “h” la cual la red no conocía:

Figura 2: Respuesta de una red de hopfield asincrónica ante un patrón desconocido.

Comparación Pseudoinversa vs Hebb

Como ya se explicó anteriormente, la pseudoinversa por lo general obtuvo mejores resultados. Por ejemplo, en la siguiente figura se puede apreciar que la psuedoinversa obtuvo un buen resultado mientras que la red normal tuvo un 2

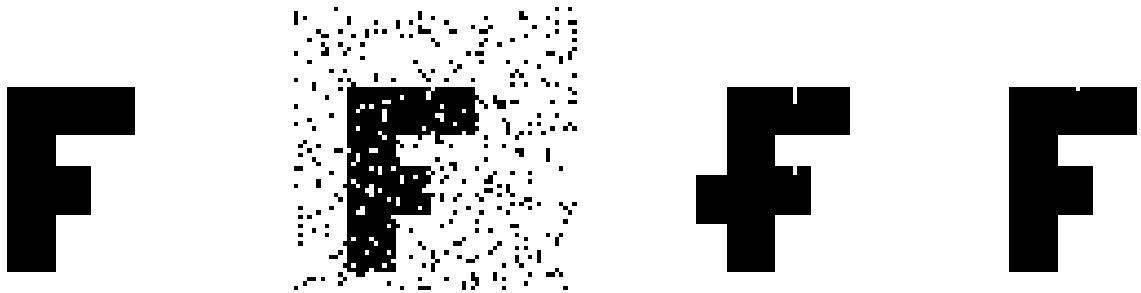

Tabla de contenidos

Cantidad de patrones almacenables

- La capacidad sugerida por *Hertz* ($0{,}138N$) no fue verificada, en la práctica fue mucho menor.
- La red logró almacenar hasta 6 o 7 imágenes.

Conclusiones

- Las Redes de Hopfield son buenas a fines teóricos pero no prácticos ya que la cantidad de memoria requerida para almacenar pequeñas imágenes es muy grande.
- La cantidad de memoria requerida por al red de Hopfield es mucho mayor al tamaño de los patrones a almacenar.
- La capacidad práctica de la red es mucho menor a la capacidad teórica, principalmente debido a la correlación entre patrones.
- Se hallaron métodos para evitar que se converja a un patrón inverso.
- El uso de la psuedoinversa obtuvo mejores resultados que la red de Hebb normal.