

# Redes de Hopfield

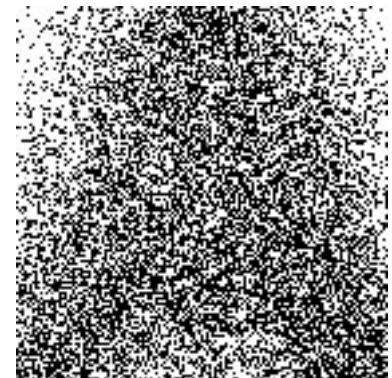
Fernando H. B. Cardoso

# Memória Associativa

- Recuperar uma memória “completa” a partir de uma pequena amostra:
  - Certos cheiros nos lembram pessoas
  - Certas canções nos lembram ocasiões
- Padrões conhecidos podem ser recuperados a partir de versões ruidosas do mesmo:
  - Que letra está escrita a seguir?



# Memória Associativa

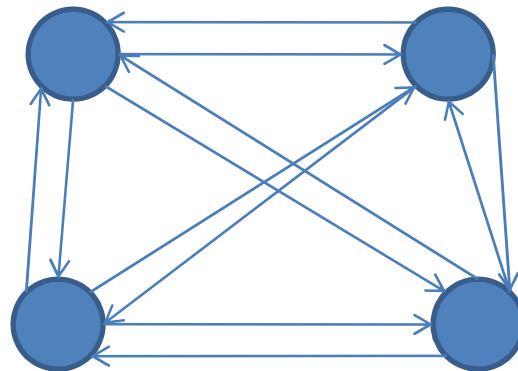


# Memória Associativa

- Coleção de dados a serem memorizados
- Redes de Hopfield são capazes de aprender padrões
- Os padrões são recuperados mesmo quando:
  - Apenas parte dele são apresentados à rede
  - Versões modificadas do mesmo são apresentadas

# Redes de Hopfield

- Modelo de memória associativa
- A saída da rede converge para um padrão aprendido
- Recorrência: a saída de um neurônio está ligada a todos os outros (atraso de tempo)



# Redes de Hopfiel

- Propriedades:
  - Unidades computacionais não-lineares
  - Simetria das conexões sinápticas ( $w_{ij} = w_{ji}$ )
  - Totalmente realimentada, exceto autoalimentação
- Neurônios formais de McCulloch Pitts
  - Saída binária (-1 .. 1)
  - Função “hard-limit”

# Aprendizado Hebbiano

- Se a ativação de um neurônio:
  - Tiver grande influência na ativação de um outro:  
REFORÇO no peso sináptico
  - Não tiver influência na ativação de outro:  
ENFRAQUECIMENTO (ou remoção) do peso sináptico.

# Armazenamento dos Padrões

- Para armazenar  $M$  padrões ( $\xi_\mu$ ,  $\mu = 1 \dots M$ )
- O  $i$ -ésimo valor de  $\xi_\mu$  é  $\xi_{\mu, i}$
- A fase de armazenamento consiste em montar uma matriz de pesos, de acordo com a seguinte equação:

$$\omega_{ji} = \frac{1}{N} \sum_{\mu=1}^M \xi_{\mu, j} \xi_{\mu, i}$$



# Armazenamento dos Padrões

- Uma vez que  $w_{ii} = 0$ , pode-se computar a matriz final de pesos de acordo com a equação ( $I$  é a matriz identidade):

$$W = \frac{1}{N} \sum_{\mu=1}^M \xi_{\mu} \xi_{\mu}^T - \frac{M}{N} I$$

# Recuperação dos Padrões

- Um padrão  $\xi_{\text{test}}$  é oferecido à rede, e sucessivas iterações ocorrem:
  - Um nó é escolhido aleatoriamente
  - Seu campo local induzido é calculado
    - Se maior que 0, liga;
    - Se menor que 0, desliga;
    - Se igual a 0, permanece
- Este processo se repete até que nenhuma mudança ocorra na rede.

# Um exemplo simples

- Rede de Hopfield com 3 nós
- Padrões a serem armazenados:
  - $(+1, -1, +1)$  e  $(-1, +1, -1)$
- Obtenção da matriz de pesos:

# Um exemplo simples

- Obtenção da matriz de pesos:

$$W = \frac{1}{N} \sum_{\mu=1}^M \xi_{\mu} \xi_{\mu}^T - \frac{M}{N} I$$

$$W = \frac{1}{3} \begin{pmatrix} +1 \\ -1 \\ +1 \end{pmatrix} \cdot (+1 \quad -1 \quad +1) + \begin{pmatrix} -1 \\ +1 \\ -1 \end{pmatrix} \cdot (-1 \quad +1 \quad -1) - 2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$W = \frac{1}{3} \begin{pmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{pmatrix} + \begin{pmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{pmatrix} - \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

$$W = \frac{1}{3} \begin{pmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{pmatrix}$$

# Um exemplo simples

- Computação do Estado (-1 -1 -1):
  - Neurônio 1:  $cli = 0$ . Estado atual (+1 +1 +1)
  - Neurônio 2:  $cli = -4$ . Estado atual (+1 -1 + 1)
  - Neurônio 3:  $cli = 4$ . Estado atual (+1 -1 +1)
  - ...

# Um exemplo simples

- Computação do Estado  $(-1 \ +1 \ +1)$ :
  - Neurônio 1:  $cli = 0$ . Estado atual  $(-1 \ +1 \ +1)$
  - Neurônio 2:  $cli = 0$ . Estado atual  $(-1 \ +1 \ +1)$
  - Neurônio 3:  $cli = -4$ . Estado atual  $(-1 \ +1 \ -1)$
  - ...

# Algumas considerações

- Quantidade limitada de padrões armazenados:

$$M_{\max} = \frac{N}{2\log_e(N)}$$

- Uma rede com 3 só garante um padrão armazenado.
- Principal limitação das redes de Hopfield.

# Cenas dos próximos capítulos

- Reconhecimento de caracteres via Perceptron
- Será que as redes de Hopfield são capazes de fazer o mesmo?
- Na próxima aula, será conduzido um experimento prático sobre as redes de Hopfield.
- Leitura recomendada: Haykin, pags 709 até 718. (O experimento vai ser mostrado aqui!)



# Estados Espúrios

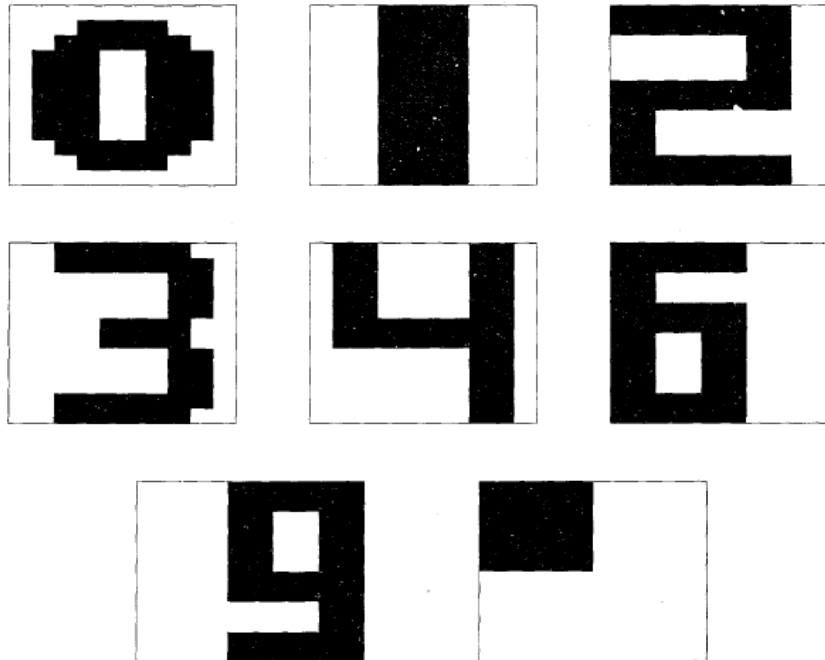
- As redes de Hopfield funcionam minimizando uma função de energia
- Todos os padrões armazenados são mínimos locais (se respeitada a equação)
- Porém, há outros mínimos locais
- As redes convergem para um mínimo local

$$E = \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N \omega_{ji} x_i x_j$$

# **EXPERIMENTOS COM REDES DE HOPFIELD**

# Descrição do Problema

- Reconhecer caracteres representados em imagens
- Número de padrões: 7



# Decisões de Projeto

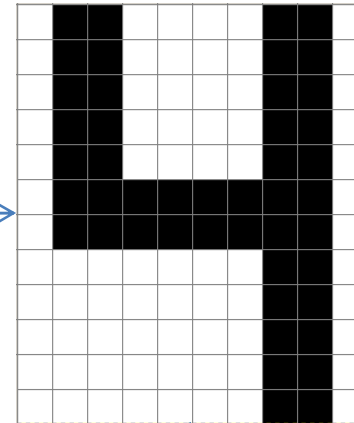
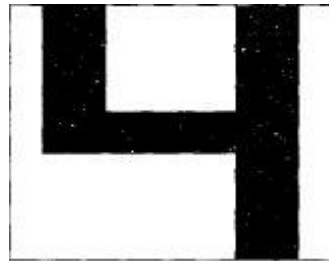
- Representar a imagem como um vetor binário (-1 e +1)

- Quantos pixels utilizar?

- 60 seriam o suficiente
- Existência de estados espúrios
- Dobrar o número de pixels?
- Desta maneira, amostram-se 10x12 pixels para cada padrão

<b>n</b>	<b>m</b>
25	3
40	5
50	6
60	7
80	9
100	10
120	12

# Obtenção dos padrões



-1, +1, +1, -1, -1, -1, -1, +1, +1, -1, -1, +1, +1, -1, -1, -1, -1, +1, +1, -1, -1,  
+1, +1, -1, -1, -1, -1, +1, +1, -1, -1, +1, +1, -1, -1, -1, -1, +1, +1, -1, -1, +1,  
+1, -1, -1, -1, -1, +1, +1, -1, -1, +1, +1, +1, +1, +1, +1, +1, +1, -1, -1, +1, +1,  
+1, +1, +1, +1, +1, +1, -1, -1, -1, -1, -1, -1, -1, -1, +1, +1, -1, -1, -1, -1, -1,  
-1, -1, -1, +1, +1, -1, -1, -1, -1, -1, -1, -1, -1, +1, +1, -1, -1, -1, -1, -1, -1,  
-1, +1, +1, -1, -1, -1, -1, -1, -1, -1, -1, +1, +1, -1

# Passos do Projeto

- Leitura dos padrões e armazenamento em memória
  - Leitura de arquivos .pgm
  - Pixels 255 (Brancos) => -1
  - Pixels 0 (Pretos) => +1
- Representação em memória: classe State
- Padrões armazenados encontram-se em um array de objetos State

# Passos do Projeto

- Criação da rede de Hopfield
  - É informada a quantidade de nós (120)
- Armazenamento dos padrões atratores
  - Para cada padrão, calcula-se a matriz intermediária
  - Somam-se todas as intermediárias e divide-se pelo número de neurônios (120)
  - Subtrai a matriz identidade, já ponderada (7/120)

# Passos do Projeto

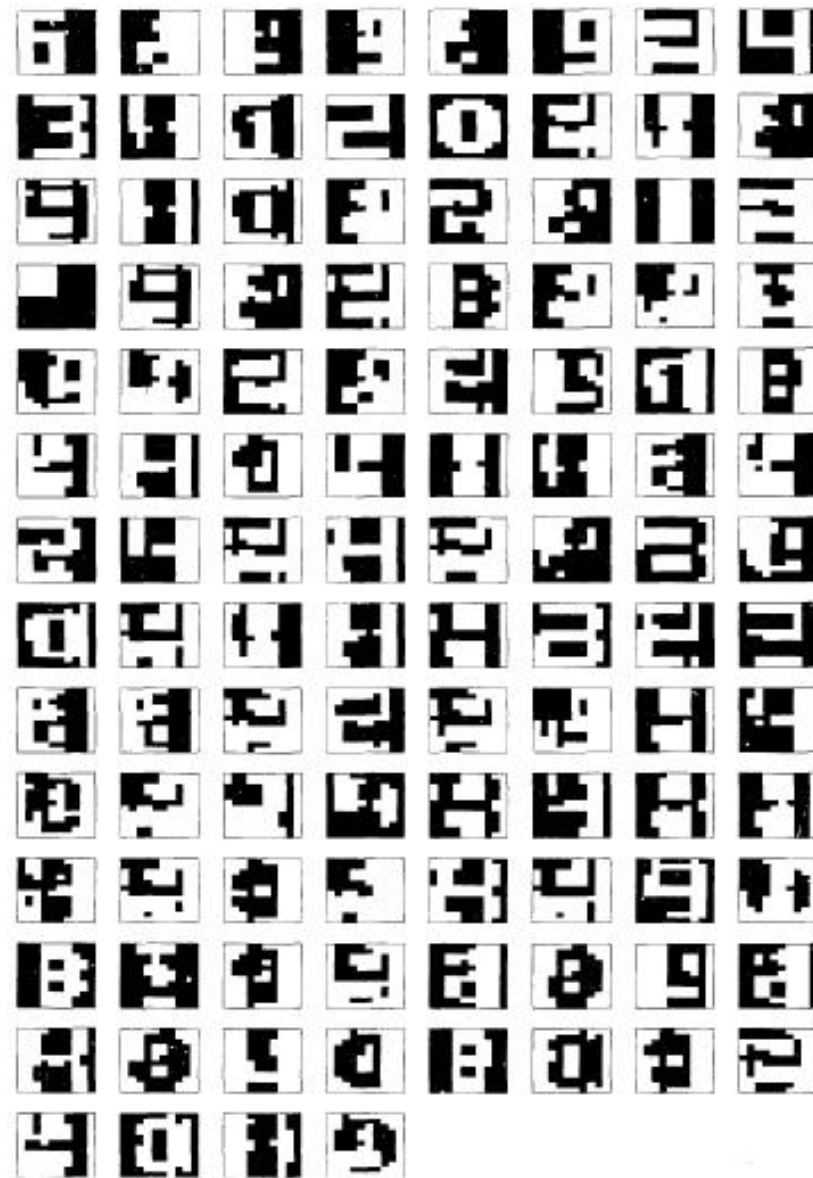
- Inicia-se a fase de recuperação
- Para cada um dos padrões armazenados:
  - Aplica-se ruído em todos os pixels, com 25% de probabilidade
  - Por ruído, entende-se a troca de valor
- Após alterar os padrões, submete-se à rede
- Funcionamento idêntico ao descrito
  - Neurônios escolhidos aleatoriamente



# Passos do Projeto

- Condições de parada:
  - Convergência para um estado armazenado
  - 1000 ativações da rede
- O número de iterações para convergência está em torno de 30
  - $120 * 25\% = 30$
- Existência de diversos estados espúrios

# Estados Espúrios



**SIMULAÇÃO**

# Conclusões

- As Redes de Hopfield são eficazes em problemas de reconhecimento de caracteres
  - MLP: mapeamento com problemas de classificação
  - Hopfield: mapeamento com problemas de recuperação de conteúdo
- Problemas:
  - Estados espúrios
  - Convergência para valores diversos