Modelo de Hopfield

Germano C. Vasconcelos
Centro de Informática - UFPE

Histórico

Em 1982 - Hopfield, Professor of Biology and Chemistry at Caltech desenvolveu um novo paradigma para Redes Neurais que impulsionou o desenvolvimento da área ...

Motivação

- Em sistemas físicos com um grande número de elementos, interações entre eles geram fenômenos coletivos estáveis ...

Isso levou Hopfield a seguinte conjectura:

- Redes de unidades de processamento que interagem entre si podem levar a fenômenos coletivos equivalentes ?

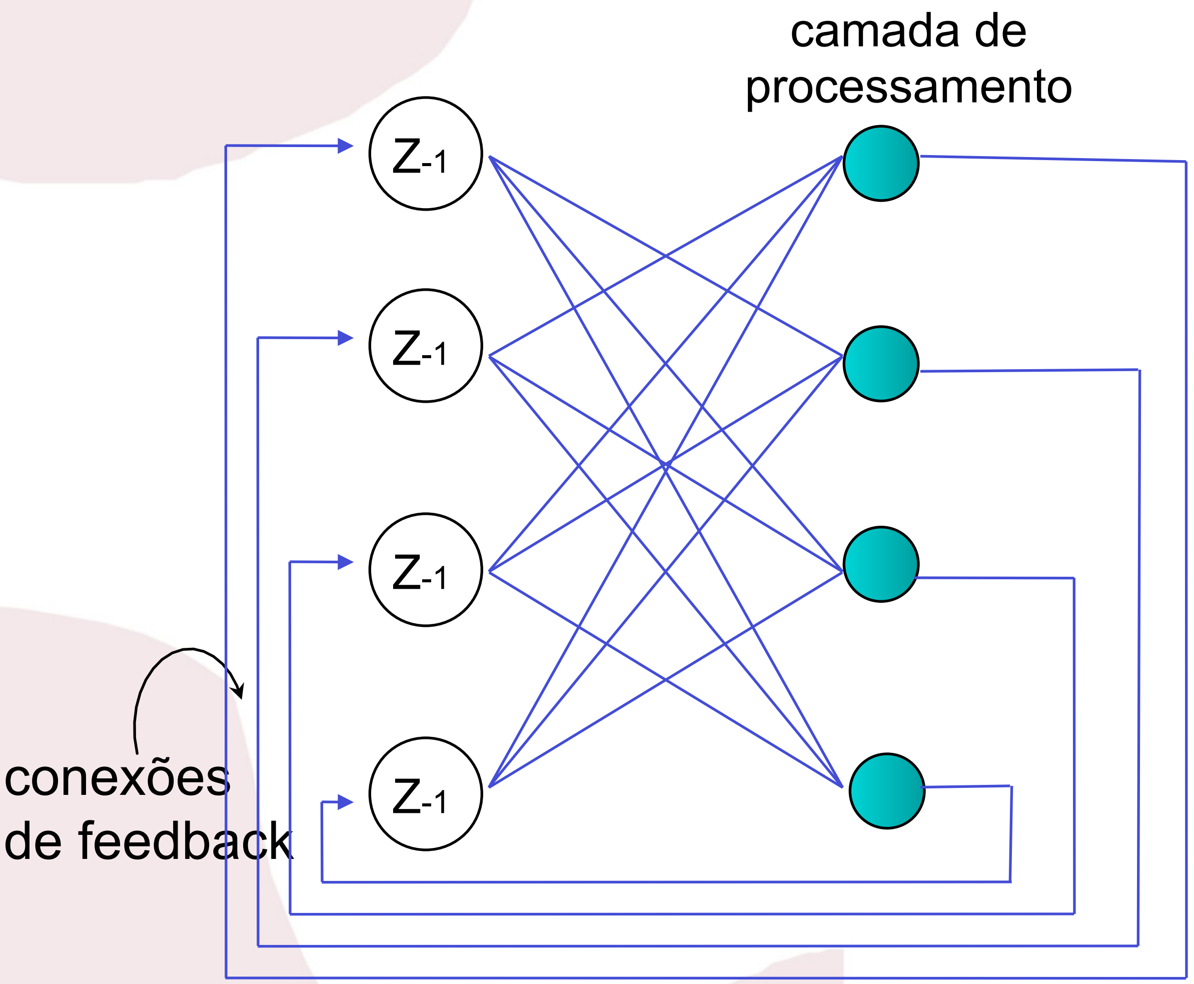
Conclusão

- Sistemas de neurônios conectados possuem estados estáveis que são atingidos quando a rede é estimulada por estados similares...

Mas qual é a grande sacada ?

Os estados podem ser obtidos através de mudança nos pesos das conexões ...

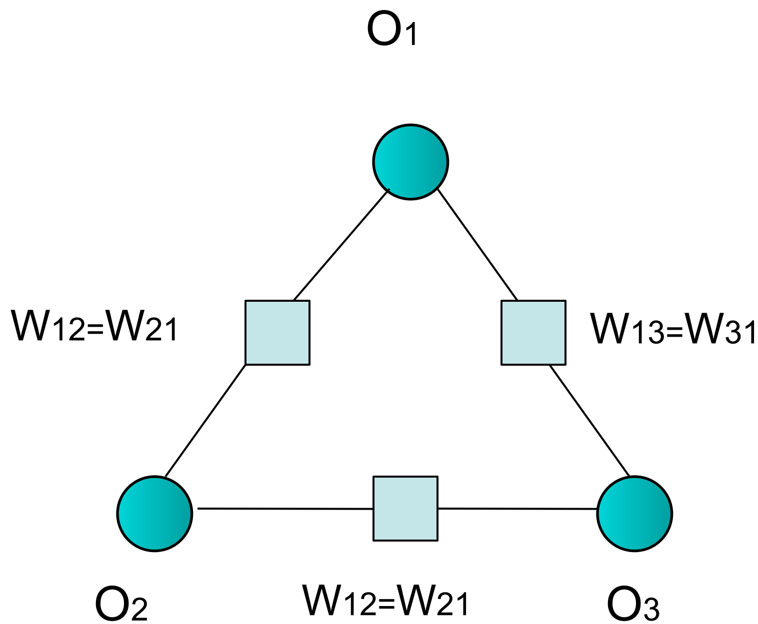
Arquitetura do Modelo



Características

- Uma única camada de unidades de processamento totalmente conectada
- Neurônios do tipo MCP
- Estrutura recorrente (com feedback)
- Unidades são ao mesmo tempo de entrada e de saída
- Funcionamento assíncrono
- Conjunto de saídas define "estado" da rede

Considere a Rede…



Operação da Rede

- Cada neurônio funciona exatamente como o MCP:

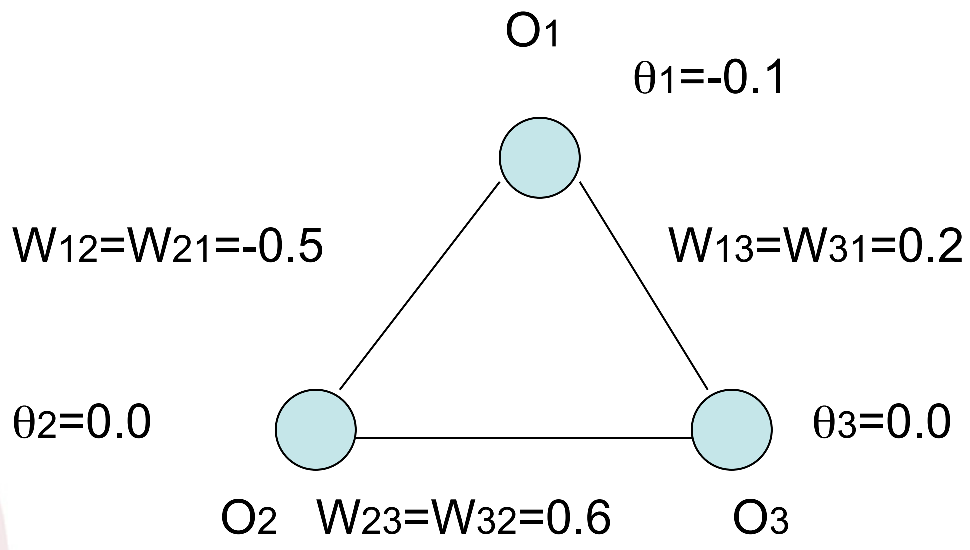
$$y = f_h\left[\sum_{i=1}^{n} w_i x_i - \theta\right]$$

com uma característica peculiar : assincronismo

Operação da Rede

- Cada neurônio pode disparar a qualquer momento com uma “taxa média” de tentativas de disparo
- Um certo no. n de tentativas/por segundo
  - em s segundos temos então em média $n.s$ disparos por neurônio
- A qualquer momento, cada neurônio tem a mesma probabilidade de disparar

Um Exemplo:


O_1
θ_1=-0.1
W_{12}=W_{21}=-0.5
W_{13}=W_{31}=0.2
θ_2=0.0
θ_3=0.0
O_2
W_{23}=W_{32}=0.6
O_3

Dado um estado O1,O2,O3 = 000 pode-se calcular o estado seguinte considerando o que aconteceria se cada um disparasse

Se neurônio 1 tentar disparar …

0x(-0.5) + 0x(0.2) = 0 > θ_1 (-0.1), resultado O1=1

O1,O2,O3 = 100

Se neurônio 2 tentar disparar …

0x(-0.5) + 0x(0.6) = 0 = θ_2 (0.0), resultado O2=0

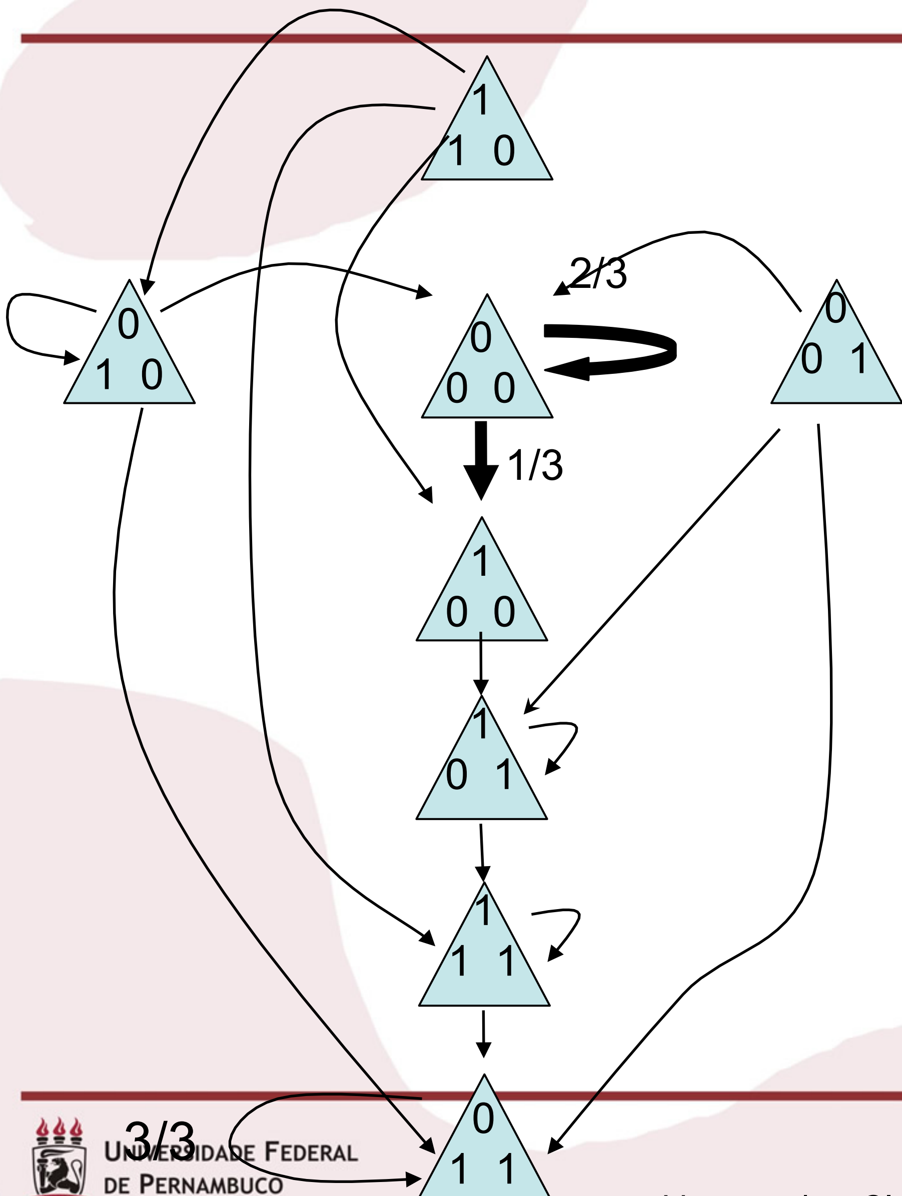
O1,O2,O3 = 000

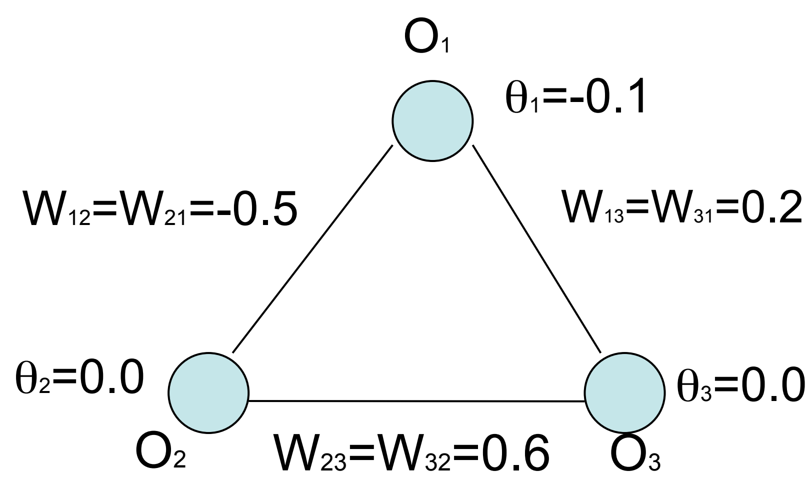
Se neurônio 3 tentar disparar …

0x(0.2) + 0x(0.6) = 0 = θ_3 (0.0), resultado O3=0

O1,O2,O3 = 000

Diagrama de Estados





O Conceito de Energia

- Uma das características mais interessantes do modelo está na associação do conceito de energia com os estados da rede …
- E, mais importante, na sua minimização como uma propriedade emergente!
- Dada uma quantidade E associada com o estado da rede …
  - E deve cair (ou permanecer como está) toda vez que um neurônio muda de estado ($Oi \rightarrow Oj$)

O Conceito de Energia

- Isso só ocorre quando :

$O_i = 0$ e $\Sigma W_{ij} O_j - \theta_i$ é positivo

então ΔO_i é positivo

ou

$O_i = 1$ e $\Sigma W_{ij} O_j - \theta_i$ é negativo

então ΔO_i é negativo

O Conceito de Energia

O que resulta no produto :

$\Delta Oi\ (\Sigma WijOj - \theta i\ )$ ser sempre positivo

Portanto, a variação na energia da rede é definida como:

$\Delta E = - \Delta Oi\ (\Sigma WijOj - \theta i\ )$

Garantindo que ΔE é sempre negativa ou nula quando um neurônio muda de estado

O Conceito de Energia

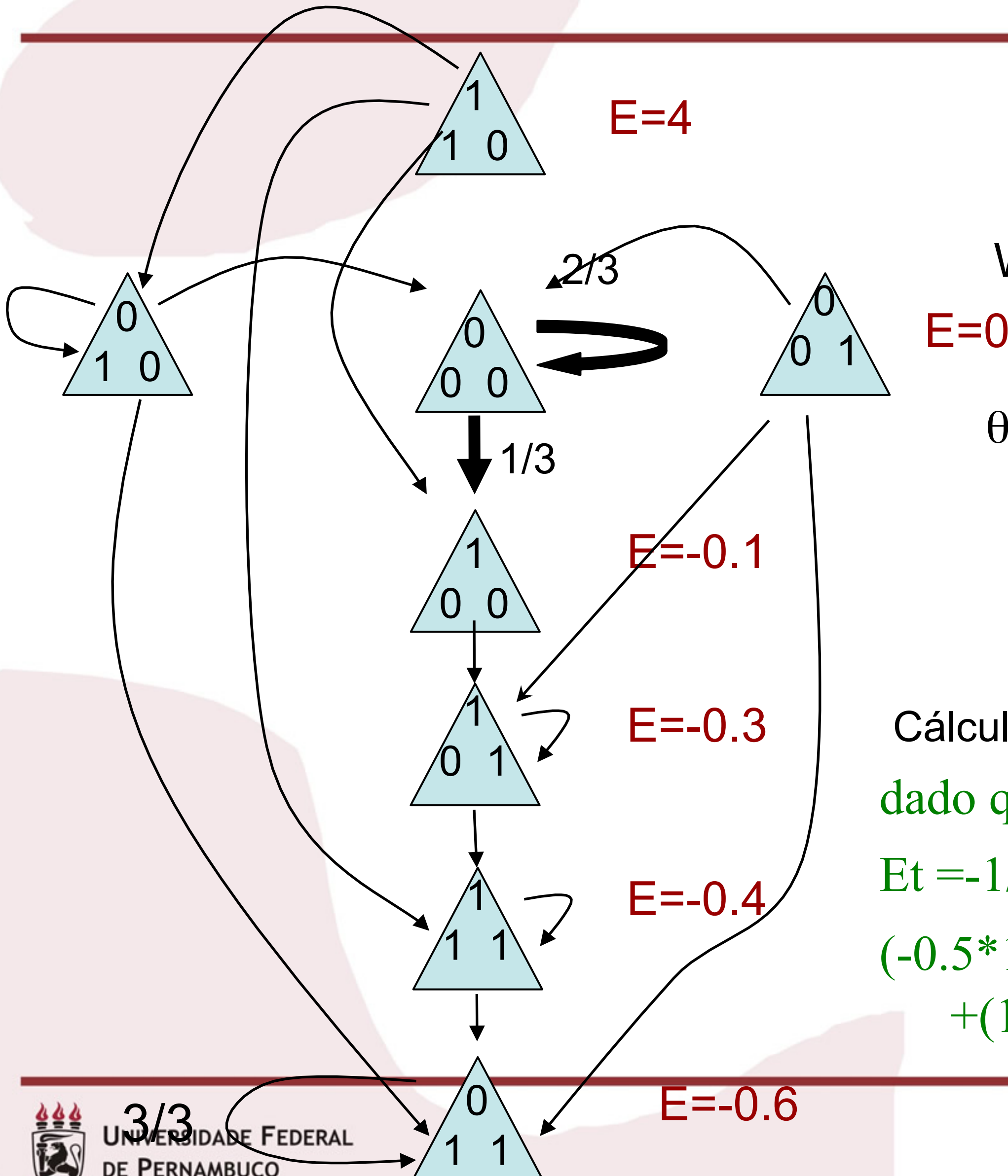
Concluindo, a energia de um nó i pode ser definida :

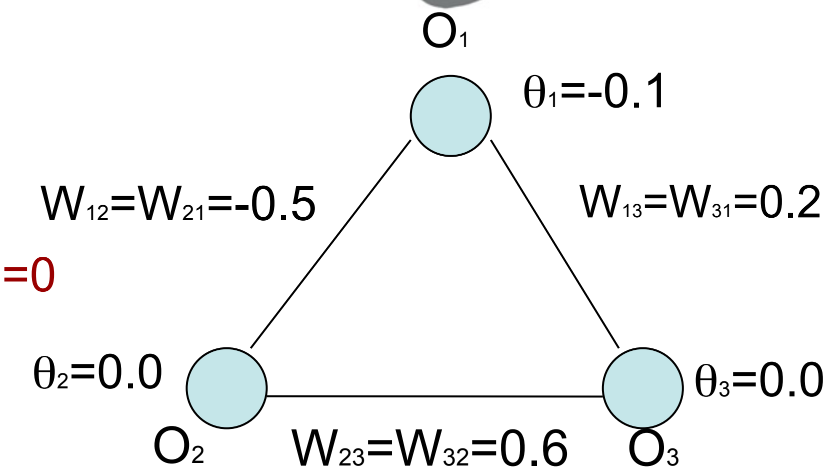
$$E = - O_i (\Sigma W_{ij} O_j - \theta_i) = - \Sigma W_{ij} O_i O_j + O_i \theta_i$$

E a energia total do sistema em um dado instante se torna:

$$E_t = - 1/2 \Sigma_i \Sigma_{j<>i} W_{ij} O_i O_j + \Sigma_i O_i \theta_i$$

Diagrama de Estados





Cálculo da energia para o estado (1 0 0)

dado que $Et = -\frac{1}{2} \sum_i \sum_{j<>i} W_{ij} O_i O_j + \sum_i O_i \theta_i$

$Et = -1/2((-0.5*0*1+0.2*0*1)+$

$(-0.5*1*0+0.6*0*0)+(0.2*1*0+0.6*0*0))$

$+(1*(-0.1)+0*(0.0)+0*(0.0)) = -0.1$

Como Definir Estados Estáveis?

"A rede só terá utilidade se houver como criar ou selecionar os estados estáveis"

Existem duas maneiras:

1. Cálculo direto das conexões
2. Treinamento da rede

Imposição de Restrições

- É alcançado através da imposição de restrições
- Para que i seja estável nenhum dos nós deve provocar mudança de estado…

$$E = - O_i \left(\Sigma W_{ij} O_j - \theta_i\right)$$

Imposição de Restrições

- Então

  se Oi é positivo $\Rightarrow$ $\Sigma W_{ij}O_j - \theta_i$ tem que ser positivo

  se Oi é negativo $\Rightarrow$ $\Sigma W_{ij}O_j - \theta_i$ tem que ser negativo

  Ex: $O_1O_2O_3 = 010$ como estável

  $O_1=0 \Rightarrow W_{12}O_2+W_{13}O_3 - \theta_1 < 0 \Rightarrow W_{12}- \theta_1 < 0$

  $O_2=1 \Rightarrow \theta_2 < 0$

  $O_3=0 \Rightarrow W_{12}- \theta_3 < 0$ (Sist. de Inequações Simultâneas)

Treinamento da Rede

Métodos para a solução de equações simultâneas ⇒ "time consuming"

Alternativa ⇒ treinamento da rede

Widrow-Hoff ⇒ $W_{ij}(t+1)=W_{ij}(t)+\eta\,[d(t)-y(t)].Oi$

(Regra Delta) ou

Produto externo ⇒ $W_{ij} = \Sigma x_{pi} x_{pj}$, para $i \neq j$ [+1,-1]

0, para $i = j$

Produto Externo $W_{ij} = \Sigma x_{pi} x_{pj}$, $i \neq j$

- Como perceptrons e MLPs, algoritmo baseia-se na minimização de uma função
- Nesse caso, a função de energia para um padrão particular *p* = $(x_0, x_1, \ldots, x_{n-1})$ (estado a ser armazenado)

$$E = -\frac{1}{2}\sum_{i}\sum_{i \neq j} w_{i,j} x_i x_j + \sum_{i} x_i T_i$$

- Para que E seja negativo, 1o *XiTi* tem que ser negativo ou igual a 0
- *Xi = -1 ou 1, Ti* teria que ter sinal oposto a *Xi.*
- Outros padrões *p* teriam valores diferentes de *Xi,* então termo com Ti pode aumentar energia. Se *Ti=0,* resolve.

Produto Externo: Treinamento

- Considerando agora o primeiro termo, separando a influência do padrão *p* na energia:

$$E = -\frac{1}{2}\sum_{i}\sum_{i \neq j} w_{i,j} x_i x_j$$

- Temos:

$$E = -\frac{1}{2}\sum_{i}\sum_{i \neq j} w'_{i,j} x_i x_j - \frac{1}{2}\sum_{i}\sum_{i \neq j} w_{i,j}^{p} x_i^{p} x_j^{p}$$

Produto Externo: Treinamento

- Primeiro termo está relacionado com contribuição de todos os demais padrões (exceto *p*) na energia e não tem como ser mexido
- Como segundo termo depende diretamente de *p* pode ser minimizado. Como é negativo, o problema se resume a maximizar:

$$\sum_{i}\sum_{i\neq j} w_{i,j}^{p}\, x_i^p x_j^p$$

- Uma forma simples de resolver é fazer $w^p_{i,j} = x_i x_j$, o que torna a expressão tão grande quanto possível já que x_i, x_j são +1 ou –1 e seu quadrado é positivo:

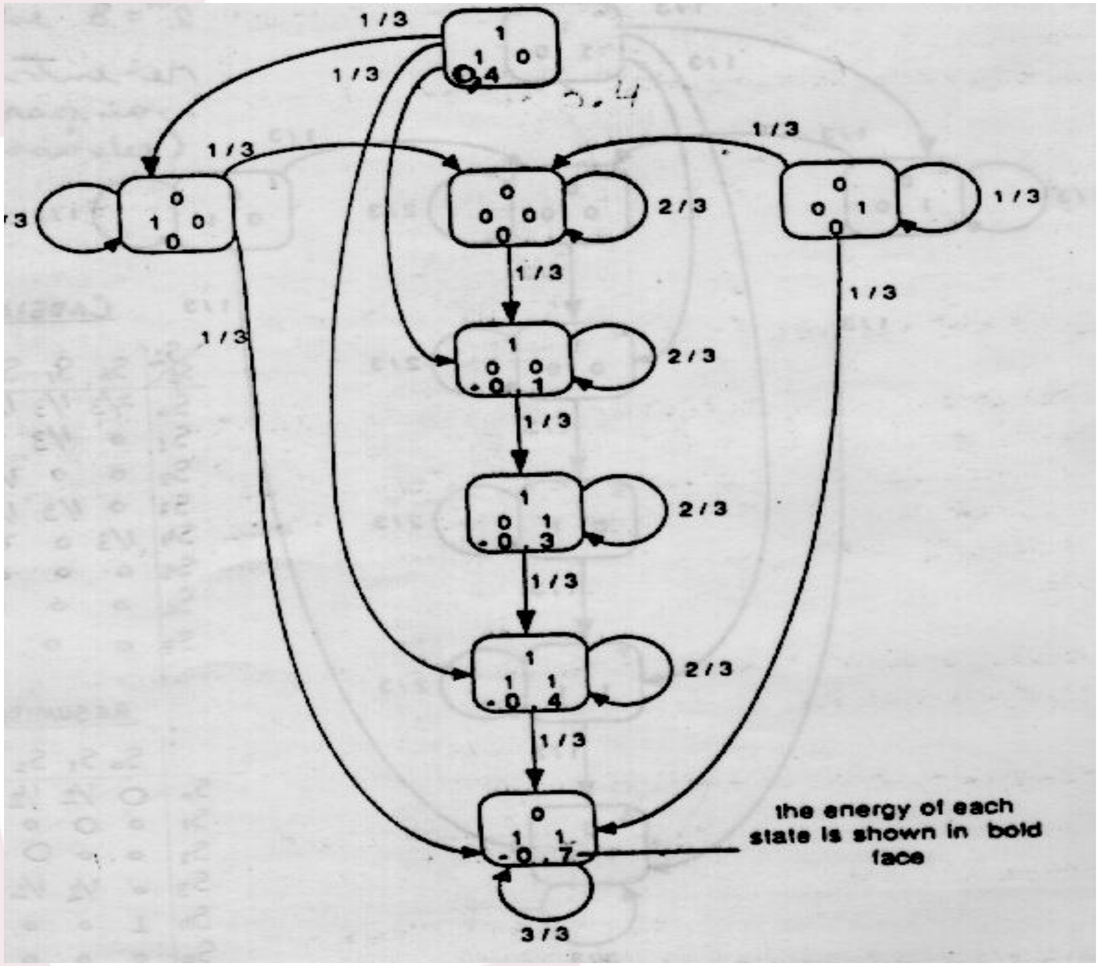
$$\sum_{i}\sum_{i\neq j} x_i^2 x_j^2$$

Produto Externo: Treinamento

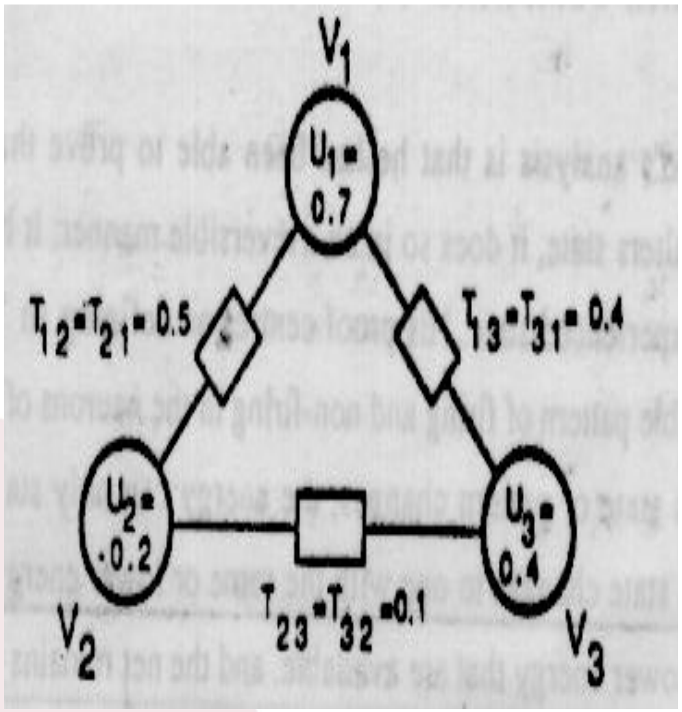
- Ou seja: fazer $\Sigma x_{pi} x_{pj}$, $i \neq j$ para todos os padrões *p* a serem armazenados

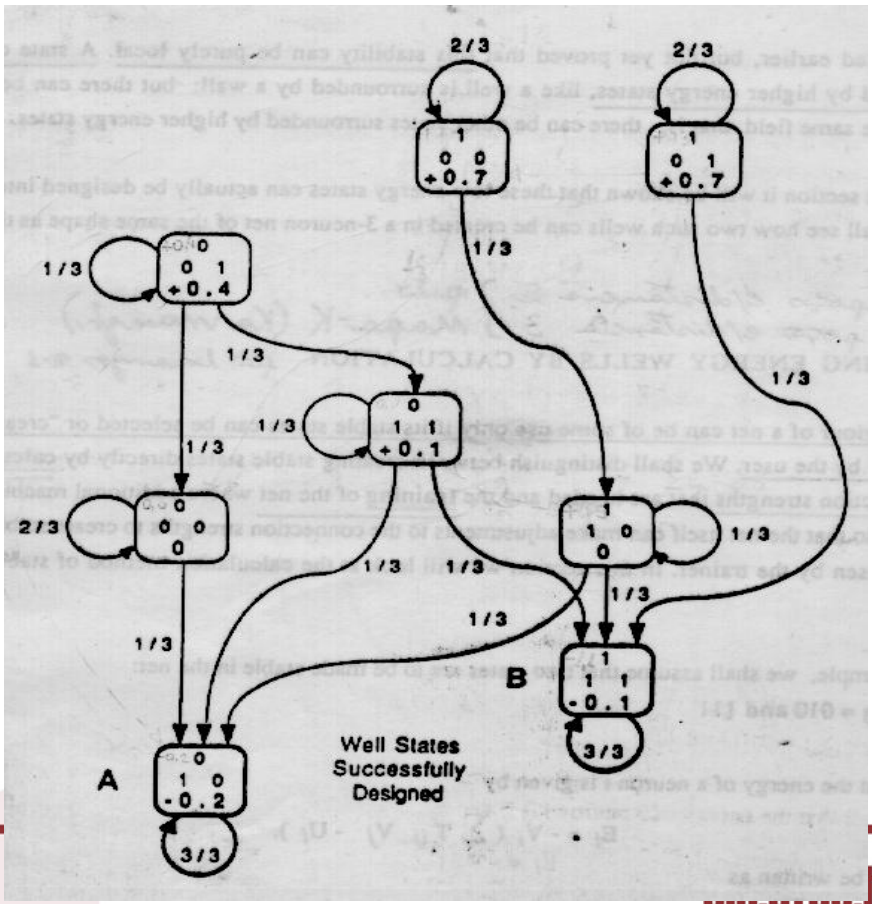
- Torna possível o treinamento da Rede de Hopfied.

Rede de Hopfield - Energia

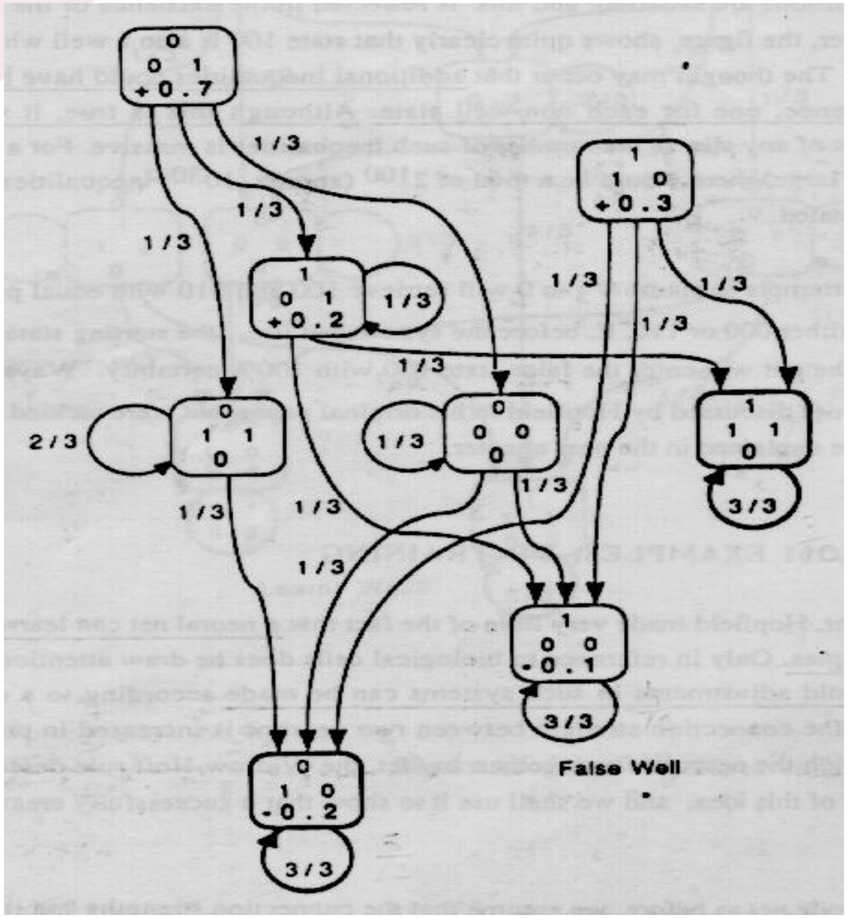


Rede de Hopfield - Aprendizagem





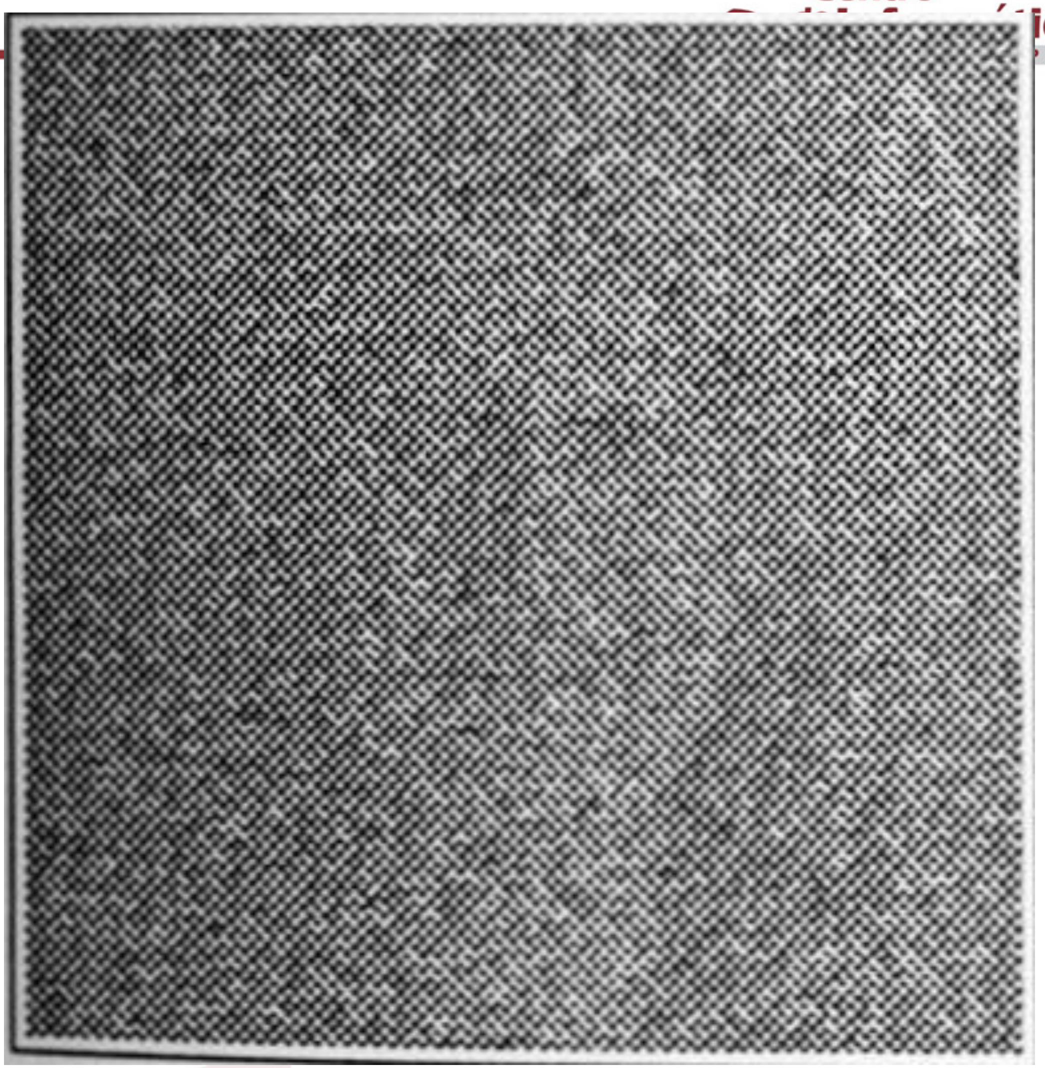
Problema: Falsos Estados Estáveis



Pattern Restoration

Pattern Completion

Exemplo de Pattern Restoration



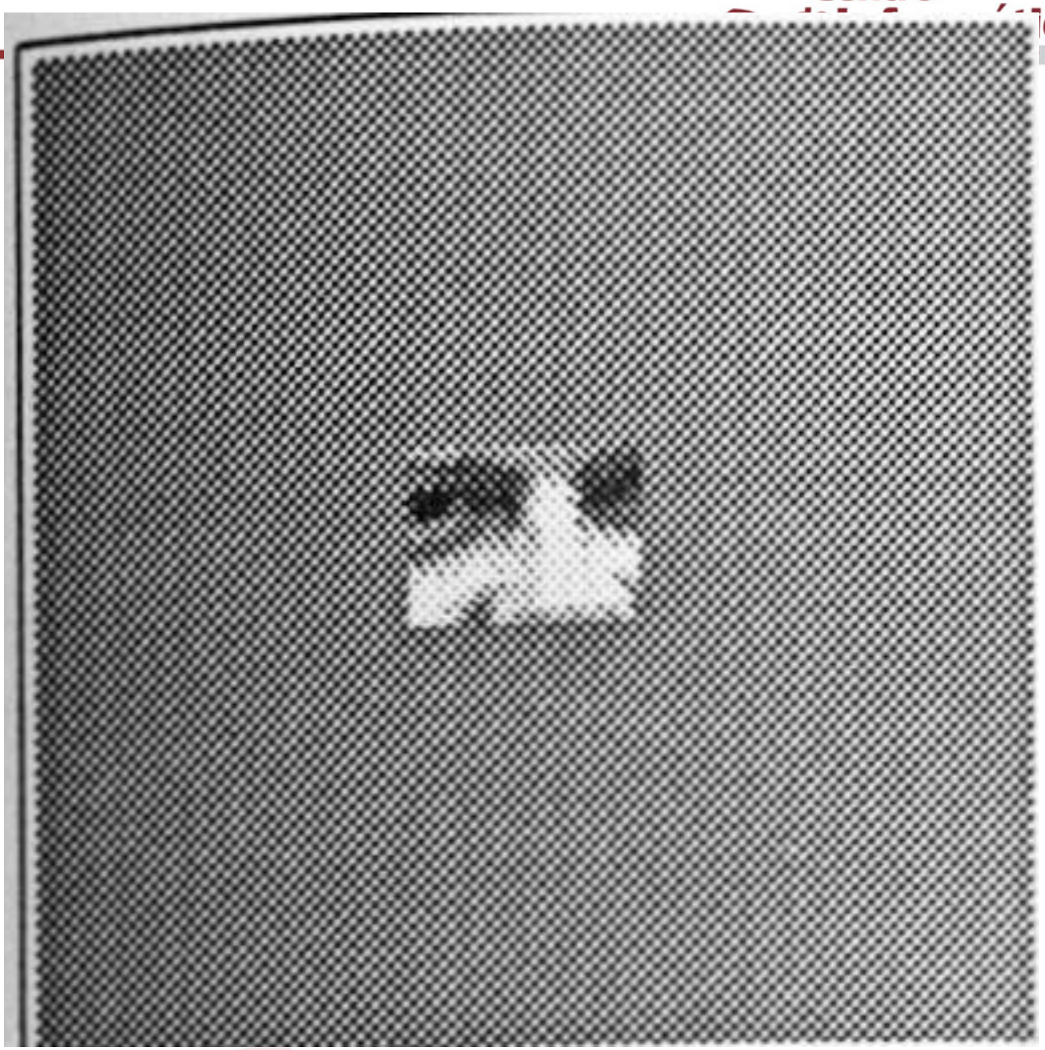
Exemplo de Pattern Restoration

Exemplo de Pattern Restoration

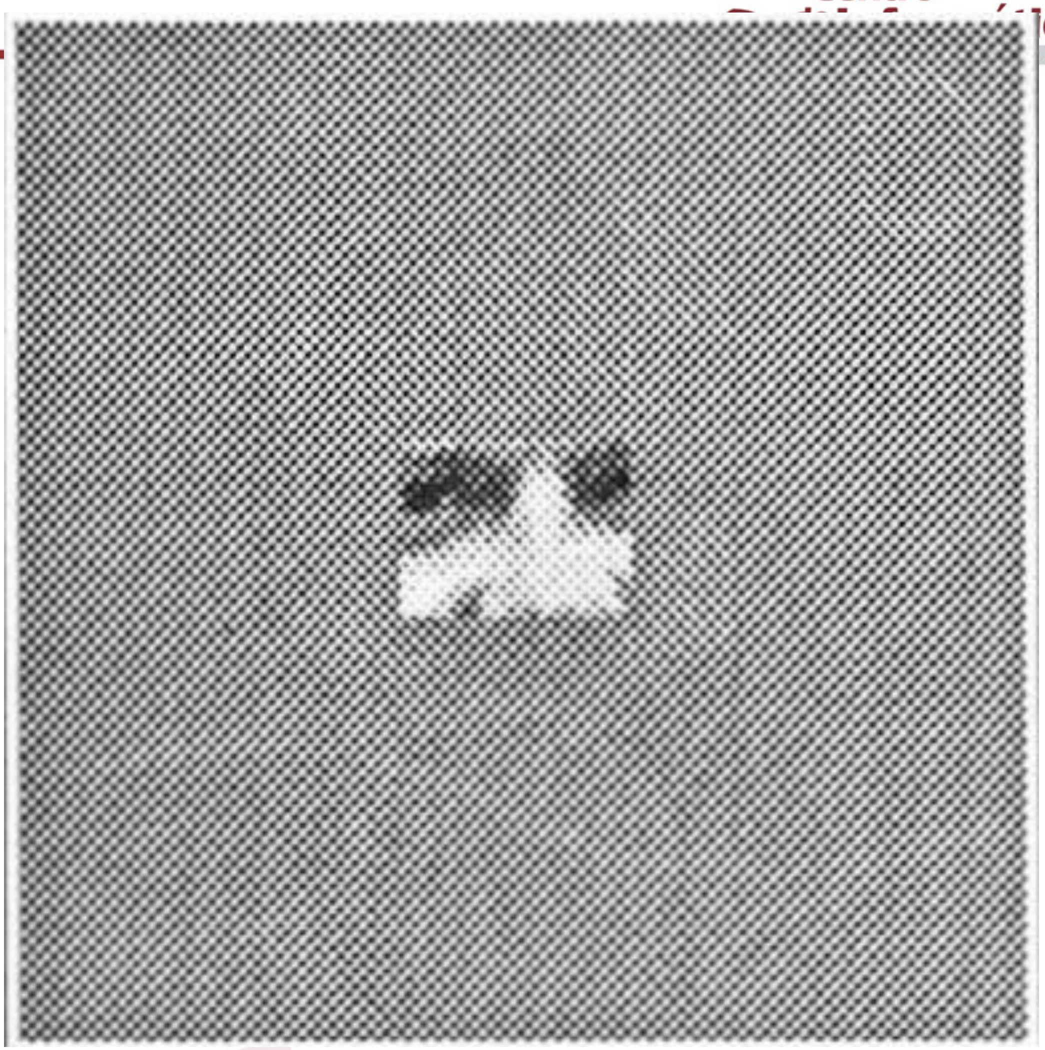
Exemplo de Pattern Restoration

Exemplo de Pattern Restoration

Exemplo de Pattern Completion



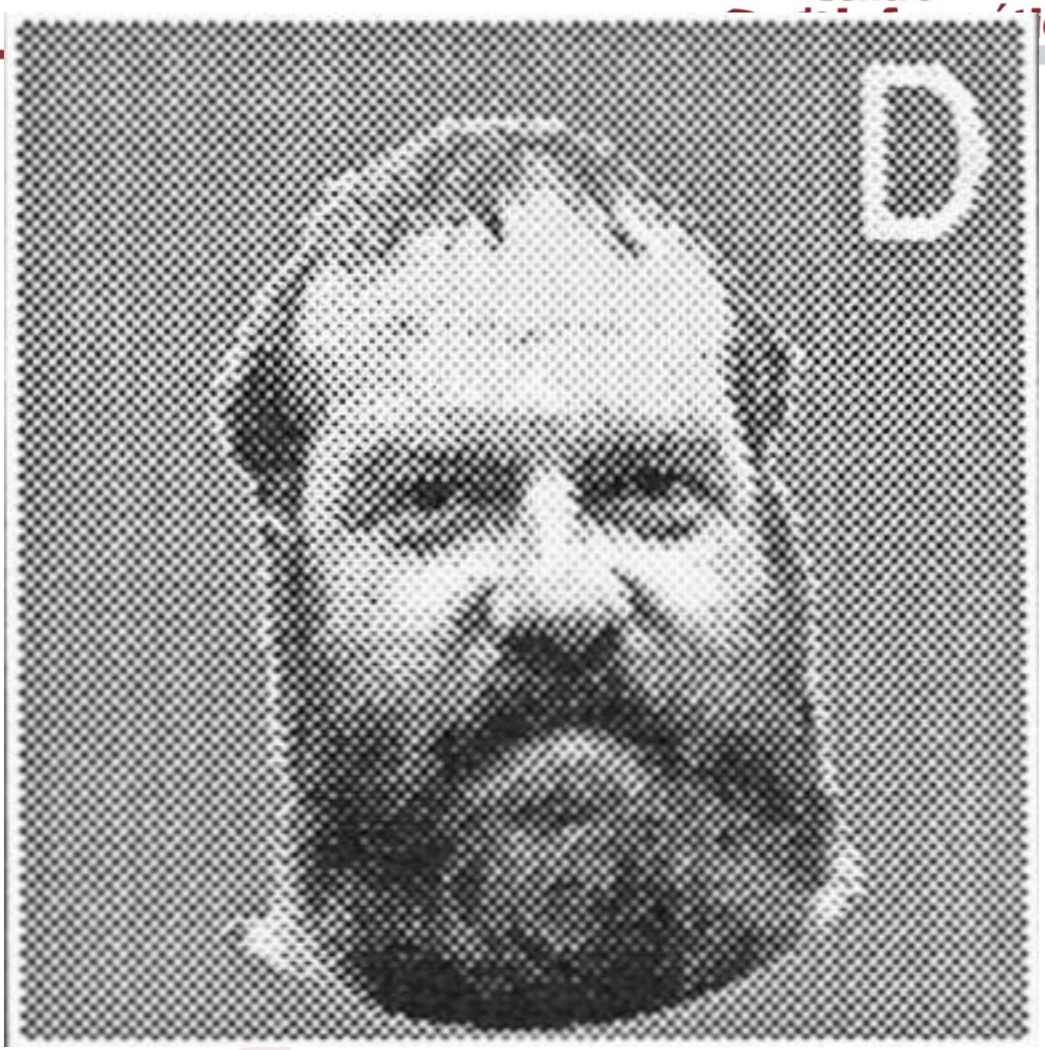
Exemplo de Pattern Completion



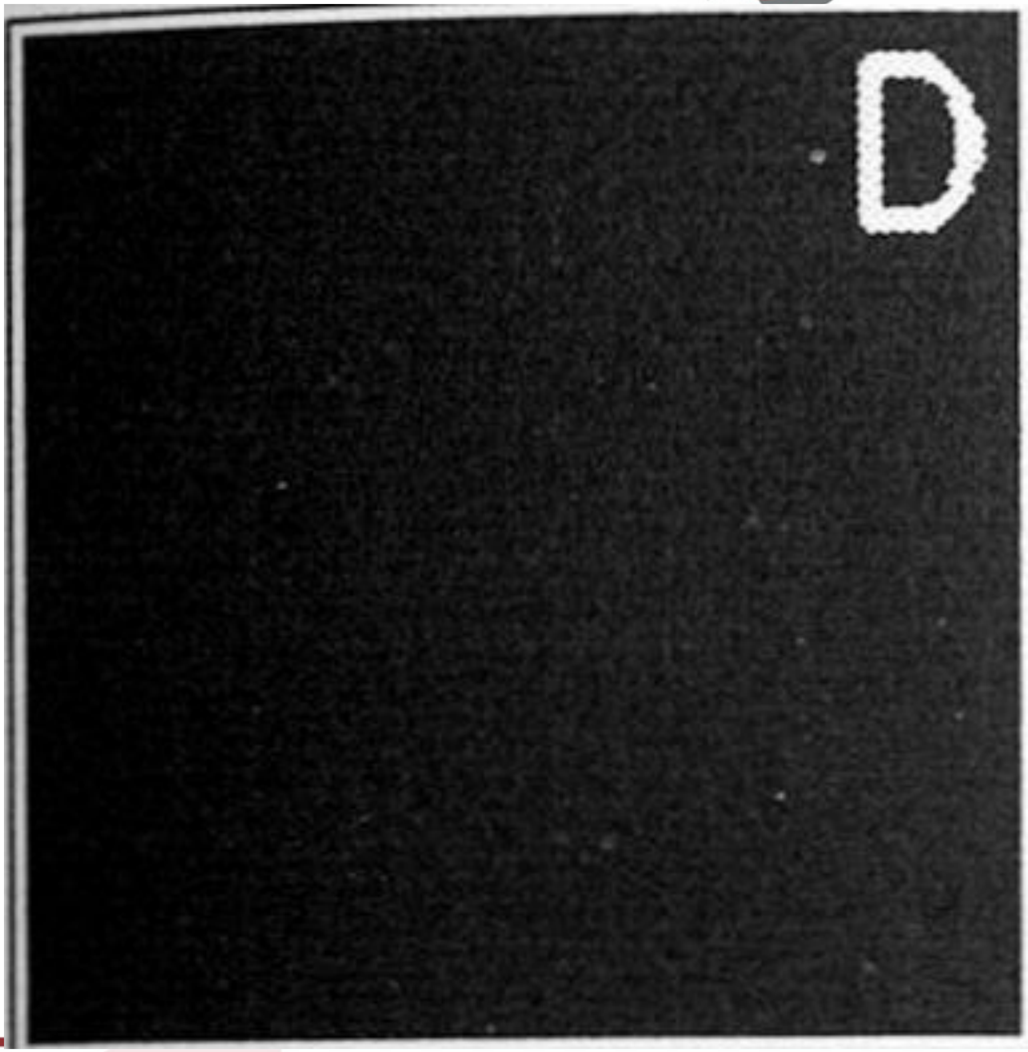
Exemplo de Pattern Completion

Exemplo de Pattern Completion

Exemplo de Pattern Completion



Exemplo de Associação

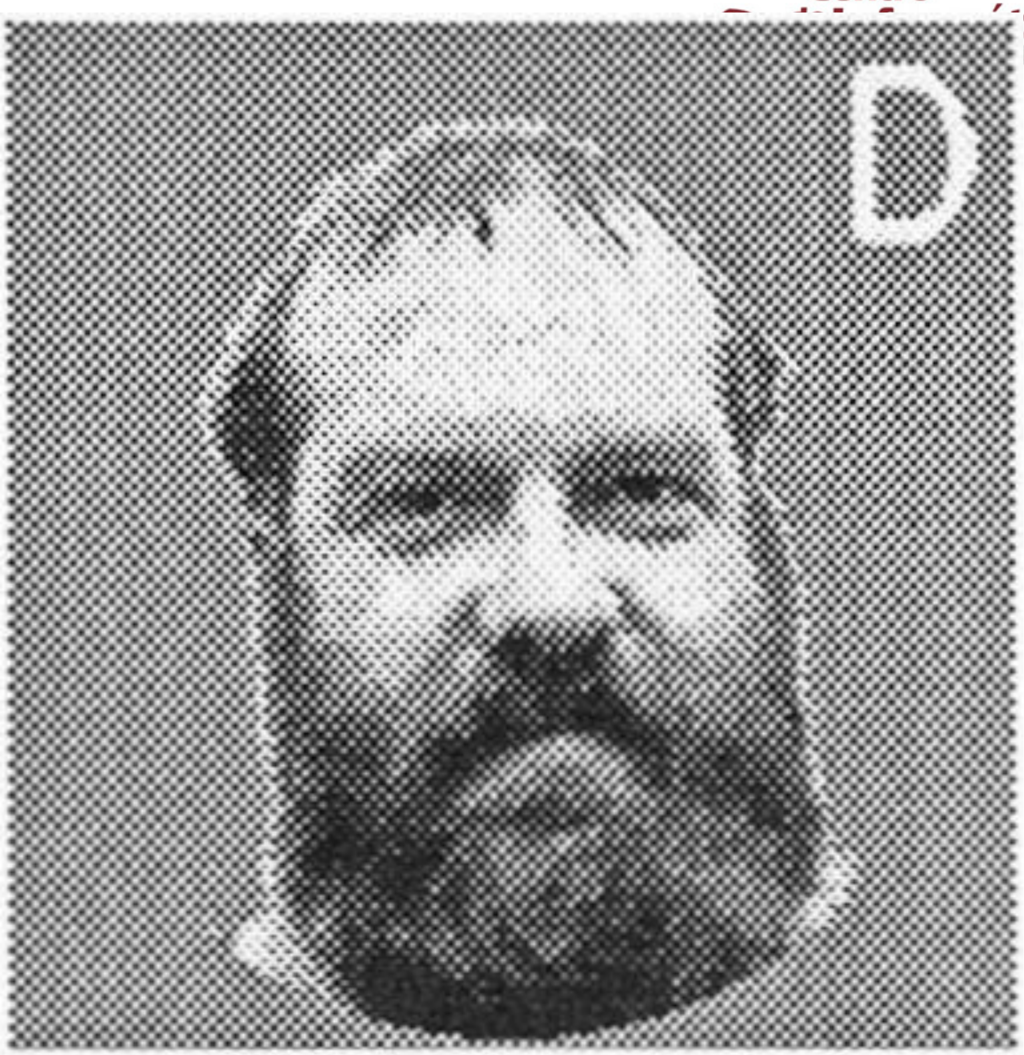


Exemplo de Associação

Exemplo de Associação

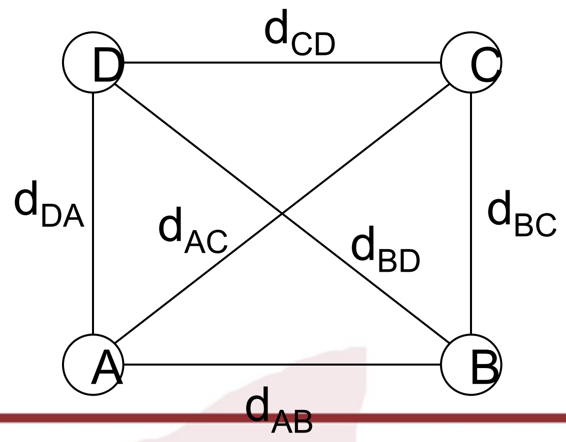
Exemplo de Associação

Exemplo de Associação



Caixeiro Viajante

- No exemplo, nós representam cidades e arestas os caminhos com as distâncias entre elas (dij)



Caixeiro Viajante

- Objetivo
  - Estabeler um rota entre as cidades, de menor distância, visitando cada cidade uma única vez
- Arquitetura da rede de Hopfield:
  - Quantos neurônios?
  - Treinamento dos pesos?

Restrições (*constraints*) definem os parâmetros

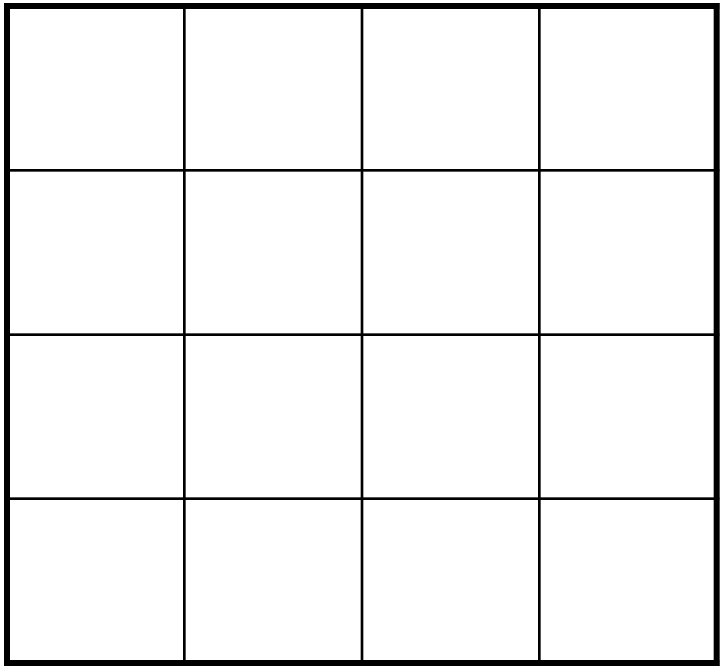
1. Para n cidades e n posições, estabelecer uma correspondência cidade-posição
   – Número de neurônios = n cidades * n posições
2. Cada cidade exatamente em 1 posição
3. Cada posição exatamente para cada 1 cidade
4. Distância total deve ser minimizada

Arquitetura

- *Matriz n * n* matrix onde linhas representam cidade e colunas posições
- *célula(i, j) = 1* se somente se cidade i^{esima} está na posição j^{esima}
- Cada célula 1 neurônio
- n^r neurônios, $O(n^4)$ conexões

pos(α)

city(i)



Restrições

1. Cada cidade em apenas 1 posição. Cada linha tem apenas um 1.

$$E_1 = \frac{A}{2} \sum_{i=1,\alpha\neq\beta}^{n} \sum_{\alpha=1} \sum_{\beta=1} x_{i\alpha} \cdot x_{i\beta}$$

Restrições

$$E_1 = \frac{A}{2} \sum_{i=1,\alpha\neq\beta}^{n} \sum_{\alpha=1} \sum_{\beta=1} x_{i\alpha} \cdot x_{i\beta}$$

Se houver situação como essa, o erro aumenta!

pos(α)

city(i)

1		1	

Restrições

2. Tem apenas 1 cidade em cada posição. Cada coluna tem apenas um 1.

$$E_2 = \frac{B}{2} \sum_{\alpha=1}^{n} \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} x_{i\alpha} \cdot x_{j\alpha}$$

Restrições

$$E_2 = \frac{B}{2}\sum_{\alpha=1}^{n}\sum_{i=1}^{n}\sum_{j=1, i\neq j}^{n} x_{i\alpha}\cdot x_{j\alpha}$$

pos(α)

city(i)

1			
1			

Se houver situação como essa, o erro aumenta!

Restrições

3. Não deve haver mais do que n 1's na matriz

$$E_3 = \frac{C}{2}\left(\sum_{i=1}^{n}\sum_{\alpha=1}^{n} x_{i\alpha} - n\right)^2$$

Restrições

- Se somarmos $E_1 + E_2 + E_3$ garantimos as restrições de ocorrências….
- Mas ainda falta?

Restrições

4. Distância mínima percorrida

$$E_4 = \frac{D}{2}[\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} d_{ij} \cdot x_{i\alpha} \cdot (x_{j,(\alpha+1)} + x_{j,(\alpha-1)})]$$

d_{ij} = distância entre cidade *i* e cidade *j*

Energia Final Minimizada

$$E = E_1 + E_2 + E_3 + E_4$$

$$E_1 = \frac{A}{2} \sum_{i=1,\alpha\neq\beta}^{n} \sum_{\alpha=1} \sum_{\beta=1} x_{i\alpha} \cdot x_{i\beta}$$

$$E_2 = \frac{B}{2} \sum_{\alpha=1}^{n} \sum_{i=1}^{n} \sum_{j=1,i\neq j}^{n} x_{i\alpha} \cdot x_{j\alpha}$$

$$E_3 = \frac{C}{2} (\sum_{i=1}^{n} \sum_{\alpha=1}^{n} x_{i\alpha} - n)^2$$

$$E_4 = \frac{D}{2} [\sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} d_{ij} \cdot x_{i\alpha} \cdot (x_{j,(\alpha+1)} + x_{j,(\alpha-1)})]$$

Regra Delta para Minimizar esta Função de Energia

Conclusões sobre o Modelo

- Forte embasamento teórico com conceitos da mecânica estatística
- Falsos estados estáveis, ou mínimos locais de energia
- Só consegue computar problema linearmente separáveis
- Capacidade de memória (armazenamento dos estados desejados)
  - N padrões de N bits
  - na prática $0.15N$
- Máquina de Boltzmann

Applets Rede de Hopfield

http://www.cbu.edu/~pong/ai/hopfield/hopfieldapplet.html

http://www.eee.metu.edu.tr/~alatan/Courses/Demo/Hopfield.htm

8. Construa o diagrama de estados para a rede de Hopfield abaixo: