

# **Computational Intelligence**

**Winter Term 2011/12** 

Prof. Dr. Günter Rudolph

Lehrstuhl für Algorithm Engineering (LS 11)

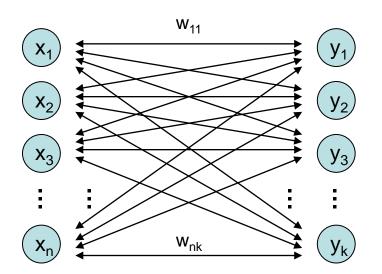
Fakultät für Informatik

**TU Dortmund** 

# **Plan for Today**

- Bidirectional Associative Memory (BAM)
  - Fixed Points
  - Concept of Energy Function
  - Stable States = Minimizers of Energy Function
- Hopfield Network
  - Convergence
  - Application to Combinatorial Optimization

### **Network Model**



x, y: row vectors

W: weight matrix

W': transpose of W

bipolar inputs  $\in \{-1,+1\}$ 

- fully connected
- bidirectional edges
- synchonized:

**start:** 
$$y^{(0)} = sgn(x^{(0)} W)$$

$$x^{(1)} = sgn(y^{(0)} W')$$

$$y^{(1)} = sgn(x^{(1)} W)$$

$$x^{(2)} = sgn(y^{(1)} W')$$

. . .

### **Fixed Points**

### **Definition**

(x, y) is *fixed point* of BAM iff 
$$y = sgn(x W)$$
 and  $x' = sgn(W y')$ .

Set 
$$W = x' y$$
. (note: x is row vector)

$$y = sgn(x W) = sgn(x(x'y)) = sgn((xx')y) = sgn(||x||^2 y) = y$$
  
> 0 (does not alter sign)

$$x' = sgn(W y') = sgn((x'y) y') = sgn((x'(y y'))) = sgn((x'(y y'))) = sgn((x'(y y'))) = x'$$
 $> 0 \text{ (does not alter sign)}$ 

**Theorem:** If W = x'y then (x,y) is fixed point of BAM.

# **Concept of Energy Function**

given: BAM with  $W = x^{2}y \Rightarrow (x,y)$  is stable state of BAM

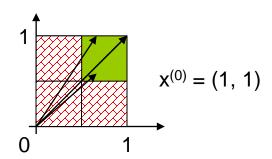
starting point x<sup>(0)</sup>

$$\Rightarrow$$
 y<sup>(0)</sup> = sgn( x<sup>(0)</sup> W )

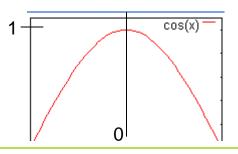
$$\Rightarrow$$
 excitation e' = W (y<sup>(0)</sup>)'

 $\Rightarrow$  if sign( e') =  $x^{(0)}$  then (  $x^{(0)}$  ,  $y^{(0)}$  ) stable state

small angle between e' and x<sup>(0)</sup>  $\leftarrow \quad \text{true if} \\ \text{e' } \textit{close} \text{ to } \mathbf{x}^{(0)}$ 



recall:  $\frac{ab'}{\|a\|\cdot\|b\|} = \cos\angle(a,b)$ 



small angle  $\alpha \Rightarrow$  large cos(  $\alpha$  )

# **Concept of Energy Function**

## required:

small angle between  $e' = W y^{(0)}$ , and  $x^{(0)}$ 

- ⇒ larger cosine of angle indicates greater similarity of vectors
- $\Rightarrow$   $\forall$ e' of equal size: try to maximize  $x^{(0)}$  e' =  $||x^{(0)}|| \cdot ||$  e  $||\cdot \cos \angle (x^{(0)}, e)$  fixed fixed  $\rightarrow$  max!
- $\Rightarrow$  maximize  $x^{(0)}$  e' =  $x^{(0)}$  W  $y^{(0)}$  '
- $\Rightarrow$  identical to minimize  $-x^{(0)}$  W  $y^{(0)}$  '

#### **Definition**

Energy function of BAM at iteration t is E( $x^{(t)}$ ,  $y^{(t)}$ ) =  $-\frac{1}{2}x^{(t)}$  W  $y^{(t)}$ .

### **Stable States**

#### **Theorem**

An asynchronous BAM with arbitrary weight matrix W reaches steady state in a finite number of updates.

#### **Proof:**

$$E(x,y) = -\frac{1}{2}xWy' = \begin{cases} -\frac{1}{2}x(Wy') = -\frac{1}{2}xb' = -\frac{1}{2}\sum_{i=1}^{n}b_{i}x_{i} \\ -\frac{1}{2}(xW)y' = -\frac{1}{2}ay' = -\frac{1}{2}\sum_{i=1}^{k}a_{i}y_{i} \end{cases}$$
 excitations

BAM asynchronous ⇒

select neuron at random from left or right layer, compute its excitation and change state if necessary (states of other neurons not affected) neuron i of left layer has changed  $\Rightarrow$  sgn(x<sub>i</sub>)  $\neq$  sgn(b<sub>i</sub>)  $\Rightarrow$  x<sub>i</sub> was updated to  $\widetilde{x}_i = -x_i$ 

$$E(x,y) - E(\tilde{x},y) = -\frac{1}{2} \underbrace{b_i (x_i - \tilde{x}_i)}_{\leq 0} > 0$$

X <sub>i</sub>	b <sub>i</sub>	x <sub>i</sub> - x̄ <sub>i</sub>
-1	> 0	< 0
+1	< 0	> 0

use analogous argumentation if neuron of right layer has changed

- ⇒ every update (change of state) decreases energy function
- ⇒ since number of different bipolar vectors is finite update stops after finite #updates

remark: dynamics of BAM get stable in local minimum of energy function!

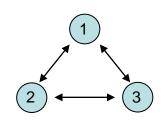
q.e.d.

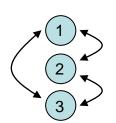
# **Hopfield Network**

special case of BAM but proposed earlier (1982)

#### characterization:

- neurons preserve state until selected at random for update
- n neurons fully connected
- symmetric weight matrix
- no self-loops (→ zero main diagonal entries)
- thresholds  $\theta$  , neuron i fires if excitations larger than  $\theta_i$





transition: select index k at random, new state is  $\tilde{x} = \text{Sgn}(xW - \theta)$  where  $\tilde{x} = (x_1, \dots, x_{k-1}, \tilde{x}_k, x_{k+1}, \dots, x_n)$ 

energy of state x is  $E(x) = -\frac{1}{2}xWx' + \theta x'$ 

### Theorem:

Hopfield network converges to local minimum of energy function after a finite number of updates.

**Proof:** assume that  $\mathbf{x_k}$  has been updated  $\Rightarrow \tilde{x}_k = -x_k$  and  $\tilde{x}_i = x_i$  for  $i \neq k$ 

$$E(x) - E(\tilde{x}) = -\frac{1}{2}xWx' + \theta x' + \frac{1}{2}\tilde{x}W\tilde{x}' - \theta \tilde{x}'$$

$$= -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j + \sum_{i=1}^{n} \theta_i x_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \tilde{x}_i \tilde{x}_j - \sum_{i=1}^{n} \theta_i \tilde{x}_i$$

$$= -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (x_i x_j - \tilde{x}_i \tilde{x}_j) + \sum_{i=1}^{n} \theta_i (x_i - \tilde{x}_i)$$

$$= -\frac{1}{2} \sum_{\substack{i=1 \ i \neq k}}^{n} \sum_{j=1}^{n} w_{ij} (x_i x_j - \tilde{x}_i \tilde{x}_j) - \frac{1}{2} \sum_{j=1}^{n} w_{kj} (x_k x_j - \tilde{x}_k \tilde{x}_j) + \theta_k (x_k - \tilde{x}_k)$$

$$= 0 \text{ if } i \neq k$$

$$= 0 \text{ if } i \neq k$$

$$= -\frac{1}{2} \sum_{\substack{i=1 \\ i \neq k}}^{n} \sum_{j=1}^{n} w_{ij} x_{i} \underbrace{\left(x_{j} - \tilde{x}_{j}\right)}_{\text{= 0 if j } \neq k} - \frac{1}{2} \sum_{\substack{j=1 \\ j \neq k}}^{n} w_{kj} x_{j} \left(x_{k} - \tilde{x}_{k}\right) + \theta_{k} \left(x_{k} - \tilde{x}_{k}\right)$$

$$= -\frac{1}{2} \sum_{\substack{i=1 \\ i \neq k}}^{n} w_{ik} \, x_i \, (x_k - \tilde{x}_k) - \frac{1}{2} \sum_{\substack{j=1 \\ j \neq k}}^{n} w_{kj} \, x_j \, (x_k - \tilde{x}_k) + \theta_k \, (x_k - \tilde{x}_k)$$

$$= -\sum_{i=1}^{n} w_{ik} x_i (x_k - \tilde{x}_k) + \theta_k (x_k - \tilde{x}_k)$$

$$= -(x_k - \tilde{x}_k) \left[ \underbrace{\sum_{i=1}^n w_{ik} x_i}_{\text{excitation } \mathbf{e}_k} - \theta_k \right] > 0$$

since:

$$\begin{array}{c|cccc} x_k & x_k - \tilde{x}_k & e_k - \theta_k & \Delta E \\ +1 & > 0 & < 0 & > 0 \\ -1 & < 0 & > 0 & > 0 \end{array}$$

> 0 if  $x_k < 0$  and vice versa

q.e.d.

# **Application to Combinatorial Optimization**

### Idea:

- transform combinatorial optimization problem as objective function with  $x \in \{-1,+1\}$  n
- rearrange objective function to look like a Hopfield energy function
- ullet extract weights W and thresholds  $\theta$  from this energy function
- ullet initialize a Hopfield net with these parameters W and ullet
- run the Hopfield net until reaching stable state (= local minimizer of energy function)
- stable state is local minimizer of combinatorial optimization problem

## **Example I: Linear Functions**

$$f(x) = \sum_{i=1}^{n} c_i x_i \rightarrow \min!$$
  $(x_i \in \{-1, +1\})$ 

Evidently: E(x) = f(x) with W = 0 and  $\theta = c$ 



choose  $x^{(0)} \in \{-1, +1\}^n$  set iteration counter t = 0 repeat choose index k at random

$$x_k^{(t+1)} = \operatorname{sgn}(x^{(t)} \cdot W_{\cdot,k} - \theta_k) = \operatorname{sgn}(x^{(t)} \cdot 0 - c_k) = -\operatorname{sgn}(c_k) = \begin{cases} -1 & \text{if } c_k > 0 \\ +1 & \text{if } c_k < 0 \end{cases}$$
 increment  $t$ 

until reaching fixed point

 $\Rightarrow$  fixed point reached after  $\Theta$ (n log n) iterations on average

## **Example II: MAXCUT**

<u>given:</u> graph with n nodes and symmetric weights  $\omega_{ij} = \omega_{ji}$ ,  $\omega_{ii} = 0$ , on edges

<u>task:</u> find a partition  $V = (V_0, V_1)$  of the nodes such that the weighted sum of edges with one endpoint in  $V_0$  and one endpoint in  $V_1$  becomes maximal

encoding: 
$$\forall i=1,...,n$$
:  $y_i = 0 \Leftrightarrow \text{node i in set } V_0$ ;  $y_i = 1 \Leftrightarrow \text{node i in set } V_1$ 

objective function: 
$$f(y) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left[ y_i \left( 1 - y_j \right) + y_j \left( 1 - y_i \right) \right] \rightarrow \max!$$

### preparations for applying Hopfield network

step 1: conversion to minimization problem

step 2: transformation of variables

step 3: transformation to "Hopfield normal form"

step 4: extract coefficients as weights and thresholds of Hopfield net

## **Example II: MAXCUT (continued)**

step 1: conversion to minimization problem

$$\Rightarrow$$
 multiply function with -1  $\Rightarrow$  E(y) = -f(y)  $\rightarrow$  min!

step 2: transformation of variables

$$\Rightarrow$$
 y<sub>i</sub> = (x<sub>i</sub>+1) / 2

$$\Rightarrow f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left[ \frac{x_i + 1}{2} \left( 1 - \frac{x_j + 1}{2} \right) + \frac{x_j + 1}{2} \left( 1 - \frac{x_i + 1}{2} \right) \right]$$

$$= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left[ 1 - x_i x_j \right]$$

$$= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} - \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} x_i x_j$$

constant value (does not affect location of optimal solution)

# **Example II: MAXCUT (continued)**

step 3: transformation to "Hopfield normal form"

$$E(x) = \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} x_i x_j = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(-\frac{1}{2} \omega_{ij}\right) x_i x_j$$

$$= -\frac{1}{2} x' W x + \theta' x$$

$$\downarrow$$

$$\downarrow$$

$$\downarrow$$

$$\downarrow$$

$$\downarrow$$

$$\downarrow$$

step 4: extract coefficients as weights and thresholds of Hopfield net

$$w_{ij} = -\frac{\omega_{ij}}{2}$$
 for  $i \neq j$ ,  $w_{ii} = 0$ ,  $\theta_i = 0$ 

remark:  $\omega_{ij}$ : weights in graph —  $w_{ij}$ : weights in Hopfield net