5) Gradient of a quadratic form:

$$E(x) = x^T W x$$

$$E(x+\Delta x) = (x+\Delta x)^T W (x+\Delta x)$$

$$= x^T W x + x^T W \Delta x + (\Delta x)^T W x + (\Delta x)^T W \Delta x$$

$$= x^T W x + (\Delta x)^T W^T x + (\Delta x)^T W x + (\Delta x)^T W \Delta x$$

$$= x^T W x + (\Delta x)^T \underbrace{(W^T + W) x}_{\nabla E} + \underbrace{(\Delta x)^T W \Delta x}_{\text{h.o.t.}}.$$

$$\Rightarrow \boxed{\nabla E = (W^T + W) x}$$

if W is symmetric $\Rightarrow$ $\boxed{\nabla E = 2Wx}$

(compare with $E = ax^2 \Rightarrow \frac{dE}{dx} = 2ax$)

6) For linear term:

$$f(x) = x^T \theta$$

$$f(x+\Delta x) = (x+\Delta x)^T \theta = x^T \theta + (\Delta x)^T \theta$$

$$\Rightarrow \boxed{\nabla f(x) = \theta}$$

Relation with Hopfield Network:

$$V(k) = \omega o(k) - \theta$$
$$o(k+1) = \Gamma(v(k))$$

$$E(o) = -\frac{1}{2} o^T \omega o + \theta^T o$$

$$\Rightarrow \nabla E = -\frac{1}{2}(\omega + \omega^T) o + \theta$$

If ω is symmetric

$$\Rightarrow \nabla E = -\omega o + \theta$$

$\Downarrow$

Hopfield eqns become:

$$V(k) = -\nabla E$$
$$o(k+1) = \Gamma(-\nabla E)$$

($\Rightarrow o(k+1) = -\Gamma(\nabla E)$ if Γ is odd symmetric such as sgn(.))

$\Rightarrow$ A gradient-like network!

expectation: Network tries to optimize E!

— Change of cost per iteration:

$$o \to o + \Delta o \quad , \quad E \to E + \Delta E$$

$$E(o) = -\frac{1}{2} o^T W o + \theta^T o$$

$$E(o+\Delta o) = -\frac{1}{2}(o+\Delta o)^T W (o+\Delta o) + \theta^T (o+\Delta o)$$

$$= \underbrace{-\frac{1}{2} o^T W o}_{1} \underbrace{- \frac{1}{2} o^T W \Delta o}_{2} \underbrace{- \frac{1}{2} \Delta o^T W o}_{2} - \underbrace{\frac{1}{2} \Delta o^T W \Delta o}_{4} + \underbrace{\theta^T o}_{1} + \underbrace{\theta^T \Delta o}_{3}$$

$$= \underbrace{E(o)}_{1} - \underbrace{(\Delta o)^T W o}_{2} + \underbrace{\theta^T \Delta o}_{3} - \underbrace{\frac{1}{2} \Delta o^T W \Delta o}_{4}$$

$$= E(o) - \Delta o^T (W o - \theta) - \frac{1}{2} \Delta o^T W \Delta o$$

$$= E(o) + \Delta o^T \cdot \nabla E - \underbrace{\frac{1}{2} (\Delta o)^T W \Delta o}_{\text{h.o.t.}}$$

$$\Rightarrow \boxed{\Delta E = E(o+\Delta o) - E(o) = + \Delta o^T \cdot \nabla E + \text{h.o.t.}}$$

Claim: In synchronous update, h.o.t. $= 0$ if $w_{ii} = 0$!

$$W = \begin{pmatrix} 0 & w_{12} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2n} \\ \vdots & & & \\ & & & 0 \end{pmatrix} \qquad \Delta o = \begin{pmatrix} 0 \\ 0 \\ \Delta o_j \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{only } j^{\text{th}} \text{ output changes!}$$

$$\Rightarrow \quad W \cdot \Delta o = \begin{pmatrix} w_{1j} \Delta o_j \\ w_{2j} \Delta o_j \\ \vdots \\ 0 \end{pmatrix} \leftarrow j^{\text{th}} \text{ component is zero.}$$

$$\Rightarrow \boxed{(\Delta o)^T W \Delta o = 0 \ !}$$

$\Rightarrow$ Hence h.o.t.'s are not effective in asynchronous update!

$$\Delta E = + \Delta o^T \cdot \nabla E + h.o.t. \qquad , \quad \boxed{v = wo - \theta = -\nabla E}$$

$$(\Delta o)^T \cdot \nabla E = -\sum_{j=1}^{n} \Delta o_j \cdot v_j$$

$$v_j \rightarrow \boxed{sgn()} \rightarrow o_j \quad \Rightarrow \quad o_j = \begin{cases} -1 & \text{if } v_j = -1 \\ +1 & \text{if } v_j = +1 \end{cases}$$

$$\text{if } v_j = -1 \Rightarrow o_j = -1 \rightarrow \Delta o_j = -1 - \underset{\text{output}}{\text{past}} = \begin{cases} 0 \\ -2 \end{cases}$$

$$\text{if } v_j = +1 \Rightarrow o_j = +1 \Rightarrow \Delta o_j = 1 - \underset{\text{output}}{\text{past}} = \begin{cases} 0 \\ 2 \end{cases}$$

$$\Rightarrow \quad \boxed{v_j \cdot \Delta o_j \geq 0 \; !}$$

$$\Rightarrow \quad \Delta E = \underbrace{-\sum_{j=1}^{n} \Delta o_j \, v_j}_{\text{negative!}} + h.o.t.$$

RESULT: If we neglect h.o.t. $\Rightarrow \Delta E \leq 0$

$\Rightarrow$ Cost decreases along the iterations.

If we use asynchronous update $\Rightarrow$ h.o.t = 0

$\Rightarrow$ cost definitely decreases!

And it decreases to a LOCAL MINIMUM of E!

Example:

$$W = \begin{pmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{pmatrix} \qquad \theta = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad O = \begin{pmatrix} O_1 \\ O_2 \\ O_3 \\ O_4 \end{pmatrix}$$

$$E = -\frac{1}{2} O^T W O = -O_1(O_2 + O_3 - O_4) - O_2(O_3 - O_4) + O_3 O_4$$

E has discrete values $\{-6, 0, 2\}$

$\min E = -6$, minimizer points: $\begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$

start with $O_0 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$ $\underline{\underline{E(O_0) = 2}}$ (not is maximum)

synchronous update:

$$O(1) = \Gamma(W O_0) = \Gamma\begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \end{pmatrix} \quad E = 2$$

$$O(2) = \Gamma(W O(1)) = \Gamma\begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

Asynchronous update (first → second - third - fourth ...)

$$\begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow[\text{update}]{\text{first}} \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{\text{second}} \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{\text{third}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\text{fourth}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

$$\xleftarrow{\text{first}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\text{second}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\text{third}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\text{fourth}} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

Becomes fixed point.

$\underline{\underline{E = -6}}$ (minimizer)

Applications: A lot of quadratic optimization problems (combinatorial) can be formulated as:

$$\min_{x_i \in \{-1,+1\}} \left(x^T A x + b^T x \right)$$

or can be transformed into this form..

⇒ Most of them are NP-complete problems (i.e. the required computation increases exponentially with dimension).

Approach: Select W and θ so that the cost of the NN is SIMILAR/EQUIVALENT to the cost to be minimized.

$$E = -\frac{1}{2} O^T W O + \theta^T O$$

⇒ $O \sim x$ → $W = -2A$, $b = \theta$

(symmetry? diagonal entries?)

⇒ Construct the Hopfield NN

⇒ Run the iterations

⇒ If converged, find the converged solution O^*.

⇒ In most of the cases O^* is a local minimum

Asynchronous update: O^* is the minimum within one Hamming distance, i.e. O^* is a corner in hypercube, and there is no adjacent corner (1 Hamming distance) which has a lower cost!

EXAMPLE FROM GRAPH THEORY:

Graph: A set of nodes (vertex), and a set of branches connected to nodes.

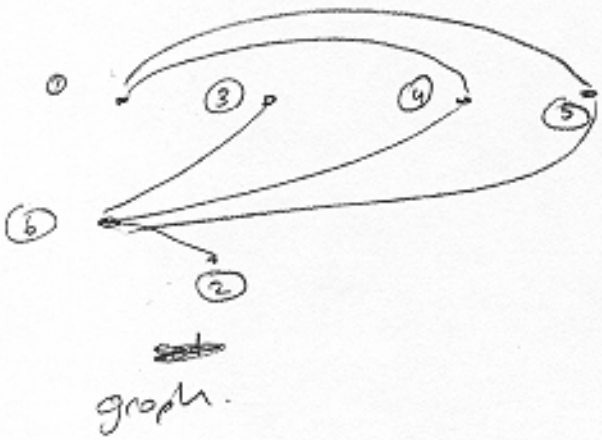
- Each branch is incident with exactly 2 nodes
- No self-loops
- no parallel branches.

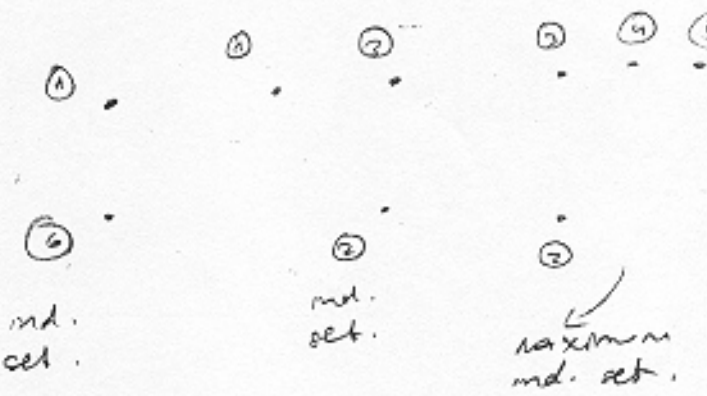
CLIQUE: A subgraph in which all nodes are (pairwise connected) with each other (complete subgraph).

maximum Clique: clique with maximum number of nodes.

Independent set: A subgraph in which nodes are pairwise not connected.

maximum independent set: An independent set with maximum number of nodes.





Adjacency matrix: Incidence matrix.

$$A = [a_{ij}] = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0 & " \quad " \quad " \quad " \text{ are not connected} \end{cases}$$

No self-loop $\Rightarrow a_{ii} = 0$; + $a_{ij} = a_{ji}$ (symmetric)

FACT: $\min_{x \in \{0,1\}^n} x^T A x - e^T x \qquad e = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

gives the max. ind. set S in the sense that $x^* = \begin{pmatrix} x_1 \\ x_2 \\ x_i \\ x_n \end{pmatrix}$

$i \in S$ node i if $x_i = 1$

node $i \notin S$ if $x_i = 0$.

$\Rightarrow$ Make it look like our system:

$$E = -\frac{1}{2} o^T W o + \theta^T o \rightarrow x^T A x - e^T x$$

$\rightarrow W = -2A \qquad \theta = -e$

$\rightarrow o(k+1) = \Gamma(W o(k) - \theta) = \Gamma\left(-2A o(k) + e\right)$

$\Gamma = \begin{cases} \Delta(\cdot) & \\ & \Delta(\cdot) \end{cases}$ instead of sgn. fct. due to (0,1) content.

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad e = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Use asynchronous update.

$$\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\0\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix}$$

$$\to \begin{pmatrix}1\\1\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \to \text{OK.} \qquad \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \to \{1,2,3\}$$

local min.

ind. set !

BUT NOT MAXIMUM!

$$\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\1\\0\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix}$$

$$\to \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix} \to \dots \qquad x^* = \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix} \to \{2,3,4,5\}$$

local min.

ind. set and MAXIMUM!

P = { class of problems for which there exists polynomial time algorithm }

NP = { " " solvable (pol. or non pol.) }

NP-complete = { a set of problems in NP such that each can be transformed by a polynomial algo }.

⇒ NP-complete problem ← maximum clique is

Application to pattern recognition:

Given patterns $y_1, y_2, \ldots, y_p$

$O(k+1) = \Gamma(WO(k) - \theta)$

Find W and θ such that

1) y_i becomes FIXED POINTS.

$$y_i = \Gamma(Wy_i - \theta) \qquad i = 1, 2, \ldots, p$$

(hence, undistorted patterns become stored patterns, retrieved correctly)

2) y_i becomes minima of

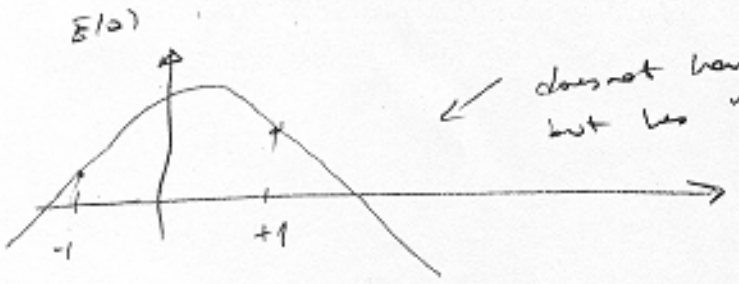
$$E = -\frac{1}{2} y_i^T W y + \theta^T y$$

3) $W = W^T$ 4) $w_{ii} = 0$.

To satisfy 2) is usually difficult. To find an efficient design which satisfies 1-4 is still a research problem.

Here 2) poses problem, because y_i is constrained!

($y_i \in \{-1, 1\}^n$!, hence minima is the sense of discrete minima!



does not have "minimum" but has "discrete minima" at $y = -1$, $y = +1$