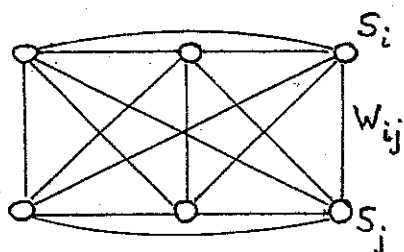


III. HOPFIELD - NETZWERKE

III.1 Hopfield-Netzwerke und Spin-Gläser



$$S_i = \pm 1$$

$$W_{ij} = W_{ji} \quad (W_{ii} = 0)$$

[meistens totale Verknüpfung]

Input: $S_i = S_i^0, \quad i = 1, \dots, N$

Dynamik:

$$S_i(t+1) = \text{sign}\left(\sum_j W_{ij} S_j(t)\right), \quad S_i(0) = S_i^0$$

→ iterative Dynamik,
synchron oder asynchron
["recurrent networks"!]

Output: $S_i = S_i^*, \quad i = 1, \dots, N$

$\{S_i^*\} = \text{stabiler Grenzzustand:}$

$$S_i^* = \text{sign}\left(\sum_j W_{ij} S_j^*\right), \quad i = 1, \dots, N$$



$$S_i^* \cdot \sum_j W_{ij} S_j^* \geq 0, \quad i = 1, \dots, N$$

→ Äquivalent mit Spin-System der statistischen Physik:

Spin-Konfiguration: $\{S_i\}, \quad S_i = \pm 1$

Wechselwirkungen: W_{ij}

Energiefunktion: $E = -\frac{1}{2} \sum_{i,j} W_{ij} S_i S_j$

lokales Feld: $h_i = \sum_j W_{ij} S_j$

"Energiefunktion" eines Hopfield-Netzwerks:

$$E = -\frac{1}{2} \sum_{i,j} W_{ij} S_i S_j = -\sum_{i>j} W_{ij} S_i S_j$$

Bei asynchroner Netzwerk-Dynamik:

$$S_i(t+1) = \text{sign}\left(\sum_j W_{ij} S_j(t)\right) = \text{sign}(h_i(t))$$

$$\begin{aligned} \rightarrow \Delta E &= E(t+1) - E(t) = -[S_i(t+1) - S_i(t)] \cdot h_i(t) \\ &= -[1 \pm 1] \cdot |h_i(t)| \end{aligned}$$

$$\rightarrow \Delta E \leq 0 \quad [\text{und } = 0 \text{ nur falls } S_i(t+1) = S_i(t)]$$

$\rightarrow$ Grenzzustand $\{S_i^*\}$ ist ein metastabiler Zustand [lokales Energieminimum!]

• Spin-Systeme:

- W_{ij} (oder deren Wahrsch.verteilung) vorgegeben
- Bestimme die metastabilen Zustände (Spin-Konfigurationen) mit tiefer Energie.

• Hopfield-Netzwerke:

- Metastabile Zustände [$\{S_i\}$ -Muster] vorgegeben
- Bestimme die W_{ij} so, dass diese Muster tatsächlich metastabile Zustände (stabile Grenzzustände) sind.

III.2 Assoziative Mustererkennung

Hopfield-Netzwerke werden hauptsächlich zur assoziativen Mustererkennung verwendet. In der Terminologie der Spinsysteme handelt es sich dabei um ein inverses Problem:

Es seien M Muster (S_i -Konfigurationen) vorgegeben:

$$\underline{S}^{\mu} = \{S_1^{\mu}, S_2^{\mu}, \dots, S_N^{\mu}\} \quad , \quad \mu = 1, \dots, M$$

Diese Muster sollen als lokale Energieminima gespeichert werden, sodass das Netzwerk aus einem Anfangszustand (gestörtes Muster) in das "ähnlichste" Muster relaxiert.

[→ Delokalisierte Informationsspeicherung!

→ Robustheit!, Fehlertoleranz!]

Das Problem besteht also darin, die Gewichte W_{ij} so zu bestimmen, dass die vorgegebenen Muster lokalen Energieminima entsprechen.

Fragen:

1) Bestimmung der W_{ij} (Lernregel) ?

2) Wieviele Muster können gespeichert werden ?

→ Kapazität: $M_{\max}(N) = ?$, $\alpha_c = \frac{M_{\max}}{N} = ?$

3) Wie stark dürfen die Inputmuster gestört sein, sodass sie immer noch erkannt werden ?

III.3 Hebb'sche Lernregel

$$W_{ij} = \sum_{\mu=1}^M S_i^{\mu} S_j^{\mu} \quad , \quad i \neq j \quad (\text{biologisch motiviert!})$$

→ Funktioniert, falls die Muster unkorreliert sind, d.h.

$S_i = \pm 1$ zufällig (Wahrsch.keit $\frac{1}{2}$) und unabhängig ausgewählt

• Approximative Analyse:

$\underline{S}^{\mu}$ ist lokales Energieminimum, falls

$$S_i^{\mu} \sum_{j=1}^N W_{ij} S_j^{\mu} \geq 0 \quad , \quad i=1, \dots, N$$

$$\text{Hebb: } \rightarrow S_i^{\mu} \sum_{\substack{j=1 \\ (j \neq i)}}^N \sum_{\mu=1}^M S_i^{\mu} S_j^{\mu} S_j^{\mu} \geq 0$$

$$\rightarrow N + \underbrace{\sum_{\mu \neq \nu}^M \sum_{j \neq i}^N S_i^{\mu} S_i^{\nu} S_j^{\mu} S_j^{\nu}}_{R_i} \geq 0$$

unkorrelierte Muster:

$$\rightarrow R_i = \sum_{k=1}^{MN} X_k \quad , \quad X_k = \begin{cases} +1 & \text{mit Wahrsch. } \frac{1}{2} \\ -1 & \text{" " " " } \frac{1}{2} \end{cases}$$

$$\rightarrow \langle R_i \rangle = 0 \quad , \quad \langle R_i^2 \rangle = MN$$

$$\rightarrow N \pm \sqrt{MN} \geq 0$$

$$\rightarrow \alpha_c = \frac{M_{\max}}{N} \lesssim 1$$

d.h. für $M \lesssim N$ (N gross!) werden die lokalen Energieminima den vorgegebenen Mustern sehr genau entsprechen.

• Genauer:

Für grosse N sind die R_i ungefähr Gauss-verteilt (zentraler Grenzwertsatz!):

$$p(R) = \frac{1}{\sqrt{2\pi MN}} e^{-R^2/2MN}$$

→ Fehlerwahrsch.keit P (= Wahrsch.keit dass ein S_i des lokalen Energieminimums nicht mit S_i^H übereinstimmt):

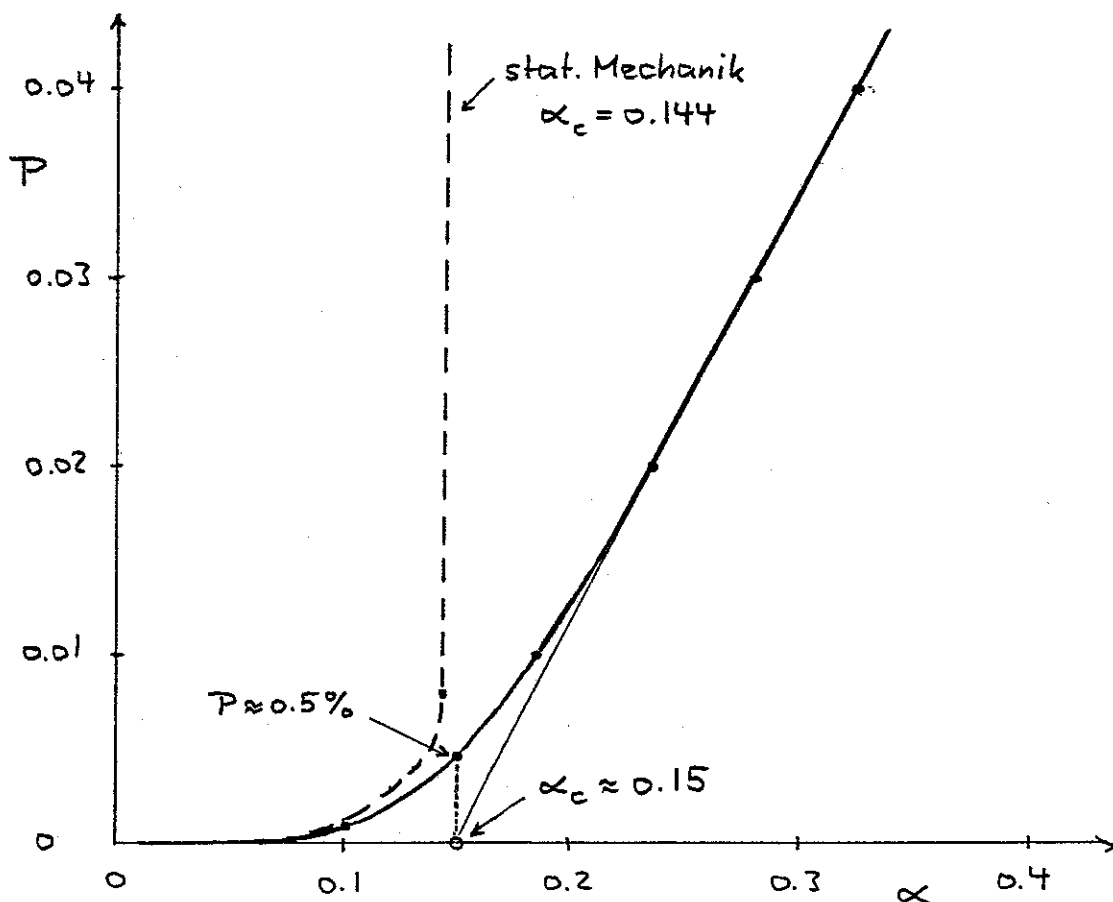
$$P = \text{Prob}(R < -N) = \text{Prob}(R > N) = \int_N^{MN} dR p(R)$$

$$= \frac{1}{\sqrt{2\pi MN}} \int_N^{MN} dR e^{-R^2/2MN} = \frac{1}{\sqrt{\pi}} \int_{\sqrt{\frac{N}{2M}}}^{\sqrt{\frac{MN}{2}}} dz e^{-z^2}$$

$$= \frac{1}{2} \underbrace{\text{erf}\left(\sqrt{\frac{MN}{2}}\right)}_{\approx 1 \text{ (} N \rightarrow \infty \text{)}} - \frac{1}{2} \text{erf}\left(\sqrt{\frac{1}{2\alpha}}\right), \quad \alpha = \frac{M}{N}$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dz e^{-z^2}$$

$$\rightarrow P = \frac{1}{2} \left(1 - \text{erf}\left(\sqrt{\frac{1}{2\alpha}}\right) \right) = \frac{1}{2} \text{erfc}\left(\sqrt{\frac{1}{2\alpha}}\right)$$



- Statistische Mechanik:

$$\alpha_c = 0.144$$

[A. Crisanti, D. J. Amit, H. Gutfreund, *Europhys. Lett.* 2, 337 (1986)]

→ $\alpha < 0.144$: Gespeicherte Muster dynamisch stabil
(kleiner Bruchteil von Fehlern)

$\alpha > 0.144$: Muster dynamisch instabil
→ Mustererkennung bricht zusammen!
(zu viele lokale Minima → "spurious minima")

- Kapazität für perfekte Speicherung:

[R. J. McEliece et al., *IEEE Trans. Information Theory*,
IT-4, 461 (1987)]

$$P = \varepsilon \rightarrow \varepsilon \approx \sqrt{\frac{\alpha}{2\pi}} e^{-1/2\alpha}, \quad \alpha = \frac{M}{N}$$

Wahrsch.keit, dass keine Fehler : $(1-\varepsilon)^N$

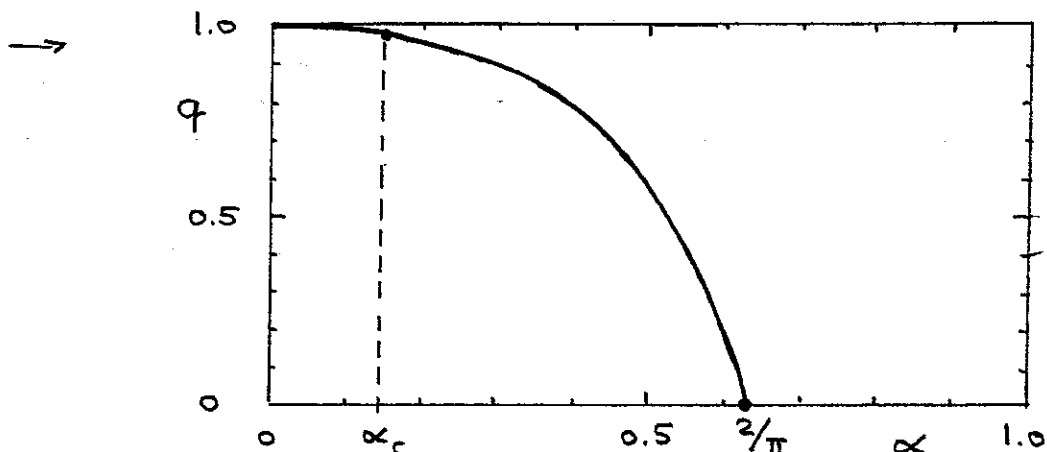
$$(1-\varepsilon)^N = \frac{1}{e} \rightarrow \varepsilon = \frac{1}{N}$$

→

$$M_c = \frac{N}{2 \ln N}$$

- Überlappung von $\underline{S}^p$ mit stabilem lokalem Minimum $\underline{S}$:

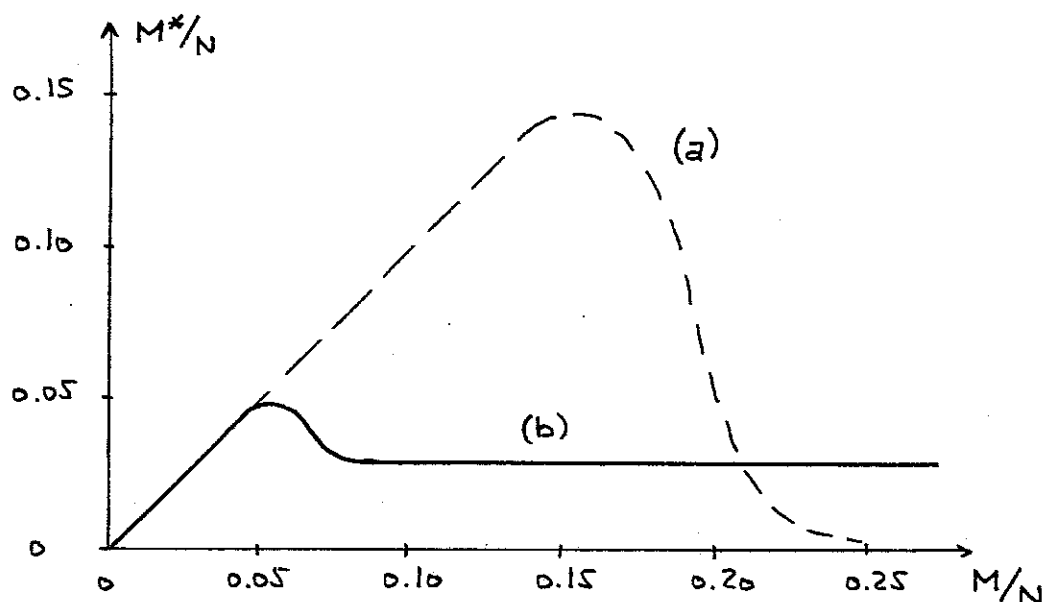
$$q = \frac{1}{N} \sum_{i=1}^N S_i^p S_i \rightarrow q = \operatorname{erf}\left(\frac{1}{\sqrt{2\alpha}} q\right)$$



• Zusammenbruch der Mustererkennung:

[J.P. Nadal et al., Europhys. Lett. 1, 535 (1986)]

[G. Parisi, J. Phys. A 19, L617 (1986)]



M = Anzahl gespeicherte Muster

M^* = Anzahl Muster, die mit 97% Genauigkeit erkannt werden

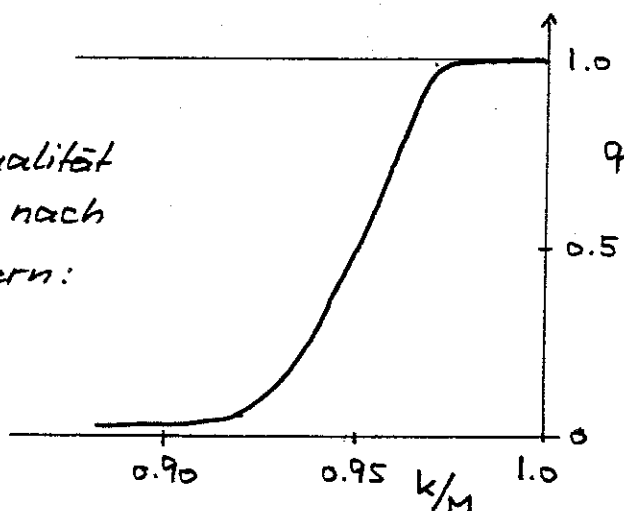
(a) $W_{ij}^{\mu} = W_{ij}^{\mu-1} + S_i^{\mu} S_j^{\mu}$ (Hebb-Regel)

(b) $-A \leq W_{ij}^{\mu} \leq +A$ (sonst wie Hebb-Regel)

→ Vergessen!

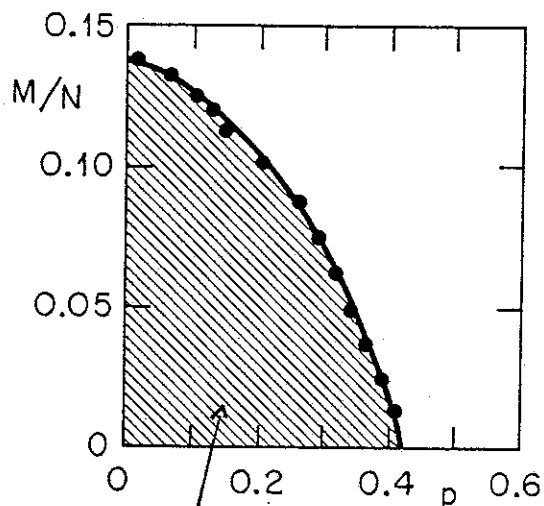
→ Zusammenbruch wird verhindert!

Wiedererkennungsqualität
des k -ten Musters nach
 M gelernten Mustern:



- Wiedererkennungsverhalten:

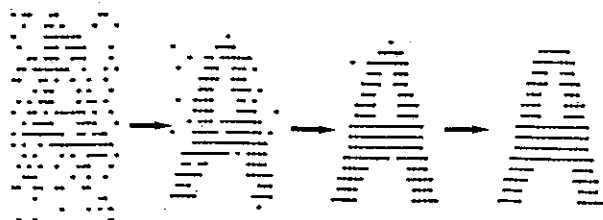
[W. Kinzel, Z. Phys. B 60, 205 (1985)]



p = Bitfehlerwahrsch.keit
im Inputmuster

$$N = 20 \times 20 = 400$$

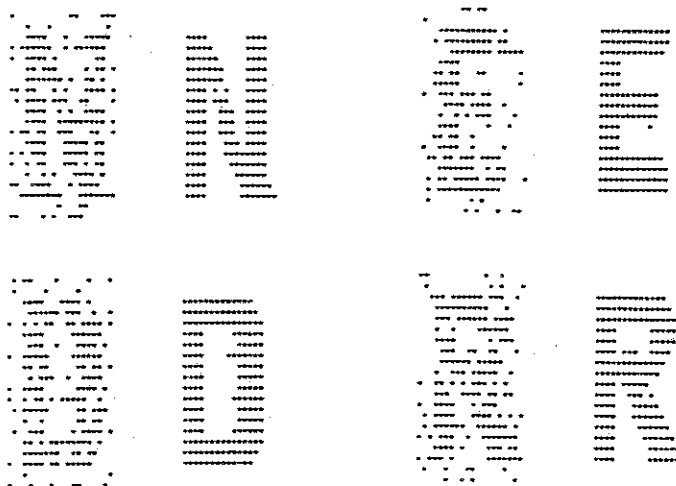
Mustererkennung mit weniger als 0.5 % Fehlern



$p=0.3$, Mustererkennung nach nur 3 (parallelen)
Iterationsschritten

Weitere Beispiele:

($p=0.2$)



III.4 Verbesserte Lernregeln

[siehe z.B. J.S. Denker, *Physica* 22D, 216 (1986)]

- Hebb'sche Lernregel:

$$W_{ij}^{\mu} = W_{ij}^{\mu-1} + \frac{1}{N} S_i^{\mu} S_j^{\mu} \quad (W_{ii} = 0 !)$$

→ schlecht für korrelierte Muster!

- "Geometrische Lernregel": ($W_{ii} \neq 0 !!$)

$$W_{ij}^{\mu} = W_{ij}^{\mu-1} + \frac{V_i^{\perp} V_j^{\perp}}{\sum_{k=1}^N (V_k^{\perp})^2}$$

wobei: $V_i^{\perp} = S_i^{\mu} - V_i^{\parallel}$, $V_i^{\parallel} = \sum_{j=1}^N W_{ij}^{\mu-1} S_j^{\mu}$

$$(W_{ij}^{\mu} = W_{ij}^{\mu-1} \text{ falls alle } V_j^{\perp} = 0 !!)$$

→ Analog zu Gram-Schmidt Orthonorm.verfahren

→ Keine Probleme mit Korrelationen!:

$$(\underline{\underline{W}}^{\mu} \underline{\underline{S}}^{\mu})_i = \sum_{j=1}^N W_{ij}^{\mu} S_j^{\mu} = S_i^{\mu} \quad \text{für } \mu=1, \dots, \nu$$

d.h. die $\underline{\underline{S}}^{\mu}$ ($\mu=1, \dots, \nu$) sind Eigenvektoren von $\underline{\underline{W}}^{\mu}$ mit Eigenwert 1 [falls die $\underline{\underline{S}}^{\mu}$ alle linear unabhängig sind!].

(alle übrigen Eigenwerte sind Null)

$$\rightarrow S_i^{\mu} \sum_{j=1}^N W_{ij}^{\mu} S_j^{\mu} = 1 > 0 \quad \text{für } \mu=1, \dots, \nu$$

→ Alle $\underline{\underline{S}}^{\mu}$ sind stabile Grenzzustände, falls sie unabhängig sind.

$$\rightarrow \alpha = \frac{\nu}{N} \leq 1 \quad (\text{da höchstens } N \text{ linear unabhängige Muster!})$$

- Maximale Kapazität eines Hopfield-Netzwerks:
(bei optimalem Lernalgorithmus)

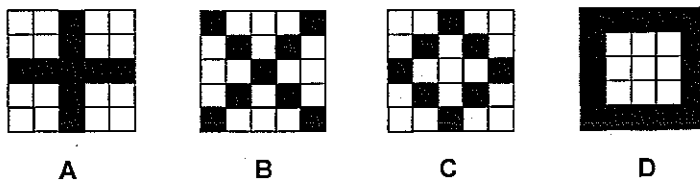
→ Statistische Mechanik:

Berechnung des Volumenanteils der W_{ij} , welche die vorgegebenen Muster als lokale Energieminima speichern. [E. Gardner, J. Phys. A 21, 257 (1988)]

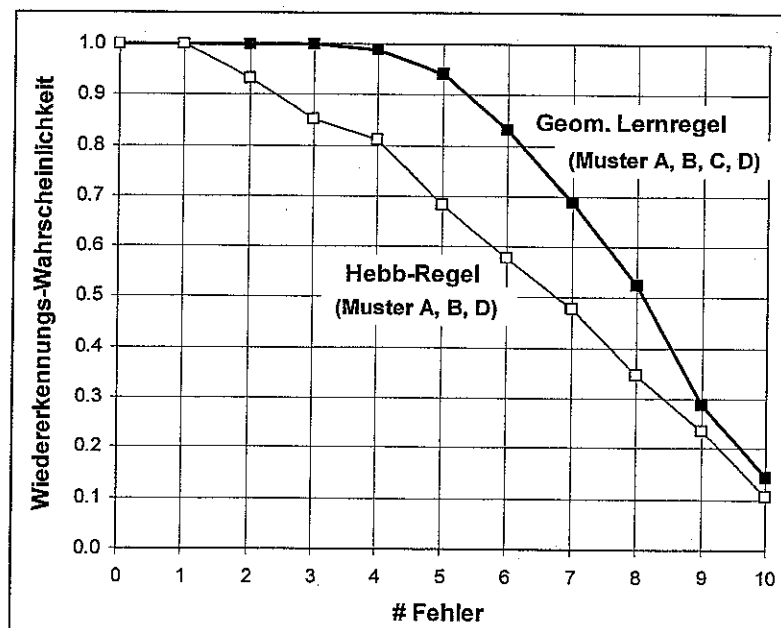
→ $\alpha_c = \frac{M_{\max}}{N} = 2$ für unkorrelierte Muster

- Beispiel für Wiedererkennungs-Verhalten:

- Zu speichernde Muster:



- Wiedererkennungsverhalten:



(mit Hebb-Regel wird Muster C nie erkannt, wenn # Fehler > 0 !)

III.5 HOPFIELD-TANK NETZWERKE ZUR LÖSUNG VON OPTIMIERUNGSPROBLEMEN

[J.J. Hopfield and D.W. Tank, Science 233, 625 (1986)]

- Verallgemeinertes Hopfield-Netzwerk:

$$S_i(t+1) = f\left(\sum_j W_{ij} S_j(t) + T_i\right)$$

→ Schwellenwerte $T_i \neq 0$

→ $f =$  oder 

→ Minimierung der Energiefunktion

$$E = -\frac{1}{2} \sum_{i,j} W_{ij} S_i S_j - \sum_i T_i S_i$$

→ LÖSUNG VON OPTIMIERUNGSPROBLEMEN!

z.B. kombinatorische Optimierungsprobleme der Form

$$F(S_1, \dots, S_N) = \min,$$

wobei die S_i nur die Werte 0 oder 1 (bzw. +1 oder -1) annehmen dürfen.

Bemerkungen:

- E muss nicht unbedingt mit F identisch sein.

Aber die Lösungen von $F = \min$ müssen auch Lösungen von $E = \min$ sein!

- $f(x) = \text{sign}(x)$: 

→ Probleme mit Nebenbedingungen

→ Probleme mit lokalen Minima

- $f(x) = \frac{1}{1+e^{-\beta x}}$ bzw. $\frac{1-e^{-\beta x}}{1+e^{-\beta x}}$: 

→ Einfachere Berücksichtigung von Nebenbedingungen (in Energiefunktion einbauen)

→ Weniger Probleme mit lokalen Minima

ANWENDUNGSBEISPIELE:

- Traveling Salesman Probleme

[z.B. J.J. Hopfield and D.W. Tank, Science 233, 625 (1986)]

- Load Flow Optimization in Power Networks

[Y. Hayashi et al., Third Symposium on Expert Systems Applications to Power Systems, Tokyo-Kobe, 1991, pp. 343-350]

[H. Mori et al., ibid., pp. 328-335]

- Optimale Team-Zusammenstellung

Beispiel:

- Zur Auswahl stehen N Personen mit individuellen Leistungsfähigkeiten R_i ($i=1, \dots, N$)
- Wenn die Personen i und j beide dem Team angehören, erhöht/vermindert sich die Leistungsfähigkeit des Teams um V_{ij} ($i, j=1, \dots, N$)
[→ "Zusammenarbeits-Matrix"]

→ Zielfunktion:

$$L = \sum_{i=1}^N R_i S_i + \frac{1}{2} \sum_{i,j=1}^N V_{ij} S_i S_j = \max$$

wobei $S_i = \begin{cases} 1 & \text{falls Person } i \text{ im Team} \\ 0 & \text{falls Person } i \text{ nicht im Team} \end{cases}$

Diese Zielfunktion hat also genau die Form einer Energiefunktion von Hopfield-Tank Netzwerken:

$$E = - \sum_{i=1}^N R_i S_i - \frac{1}{2} \sum_{i,j=1}^N V_{ij} S_i S_j = \min$$

d.h.: Schwellen $T_i = R_i$, Gewichte $W_{ij} = V_{ij}$

(Aktivierungsfkt. der Neuronen: $f(x) = \frac{1}{1+e^{-\beta x}}$)

- Bei anderen Problemen (z.B. Traveling Salesman) ist die Zielfunktion linear, und die Nicht-linearität ergibt sich aus Nebenbedingungen.

Solche Probleme lassen sich mit Hopfield-Tank Netzwerken lösen, wenn die Nebenbedingungen durch einen quadratischen Term in der Zielfunktion berücksichtigt werden können.

(siehe Beispiel "Optimale Aufgabenzuteilung")

BEISPIEL : OPTIMALE AUFGABENZUTEILUNG

[D.W. Tank and J.J. Hopfield,
Scientific American 257, pp. 62-71 (Dec. 1987)]

Effizienztabelle R_{ij} :

		Aufgaben j					
		1	2	3	4	5	6
Personen i	1	10	5	4	6	5	1
	2	6	4	9	7	3	2
	3	1	8	3	6	4	6
	4	5	3	7	2	1	4
	5	3	2	5	6	8	7
	6	7	6	4	1	3	2



R_{ij} = Effizienz der Person i für Aufgabe j

$$\rightarrow F = \sum_{i=1}^6 \sum_{j=1}^6 R_{ij} S_{ij} = \max \quad [\text{bzw. } -F = \min]$$

wobei:

$$S_{ij} = \begin{cases} 1 & \text{falls Person } i \text{ der Aufgabe } j \text{ zugeordnet} \\ 0 & \text{sonst} \end{cases}$$

Nebenbedingungen:

$$\sum_{i=1}^6 S_{ij} = 1$$

$$\sum_{j=1}^6 S_{ij} = 1$$

→ Hopfield-Tank Netzwerk:

$$E = -\frac{1}{2} \sum_{ij, ke} W_{ij, ke} S_{ij} S_{ke} - \sum_{ij} T_{ij} S_{ij} = \min$$

$$\rightarrow T_{ij} = R_{ij}$$

$$W_{ij, kj} = -C \quad (k \neq i) \quad \leftarrow \text{Negative Gewichte verhindern Verletzung der Nebenbedingungen}$$

$$W_{ij, ik} = -C \quad (k \neq j) \quad \leftarrow$$

$$[\text{übrige } W_{ij, ke} = 0]$$

$$\text{Aktivierungsfunktion der Neuronen: } f(x) = \frac{1}{1 + e^{-\beta x}}$$

Resultate [$5 \leq C \leq 10$, $\beta = 1$ oder 2]:

1	0	0	0	0	0
0	0	0	1	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0.98

$$F = 42$$

1	0	0	0.02	0.01	0
0	0	0.27	1	0	0
0	0.12	0	0.01	0	0.99
0.01	0	1	0	0	0
0	0	0	0.01	1	0.05
0.05	0.99	0	0	0	0

$$F = 44$$

(globales Maximum!)