

IV.8 Ableitungen für **gestörte** Verbindungselemente

Verbindungsgewichte können nicht nur ausfallen, sondern auch durchschalten, d.h. $w_{ij}(t) = 0$ oder $w_{ij}(t) = 1$, $\forall t \in \{0.. \infty\}$. Es sei $p_0 := P(w_{ij}(t)=0 | q)$ die bedingte Wahrscheinlichkeit **dafür**, daß das Verbindungsgewicht i des Prozessorelementes j ausfällt, unter der Bedingung, daß das Verbindungsgewicht (mit der Wahrscheinlichkeit q) gestört ist und $p_1 := P(w_{ij}(t)=1 | q)$ die bedingte Wahrscheinlichkeit **dafür**, daß das Verbindungsgewicht i des Prozessorelementes j **durchschaltet**, unter der Bedingung, daß das Verbindungsgewicht gestört ist. Die Wahrscheinlichkeit, daß ein Verbindungsgewicht ausfällt, ist dann $q * p_0$ und die Wahrscheinlichkeit für das Durchschalten eines Verbindungselementes $q * p_1$. Hierbei gilt $p_0 + p_1 = 1$, d.h. tritt ein Fehler auf, so schaltet das Element durch oder fällt aus. Die Wahrscheinlichkeit für die Aktivierung einer Verbindung durch das Lernen der Muster, ist analog zu Abschnitt IV.7 unabhängig vom Grad der Störung q :

$$P_{on} = 1 - \left(1 - \frac{k * l}{m * n}\right)^z.$$

Es sei $f := 1 - q$ die Wahrscheinlichkeit dafür, daß ein Verbindungselement einwandfrei funktioniert. Damit ergibt sich die Wahrscheinlichkeit, daß ein Knoten aktiviert ist, nachdem alle z Muster angelehrt wurden, als Summe der beiden einzelnen Wahrscheinlichkeiten:

$$P_{on,f} = \left(1 - \left(1 - \frac{k * l}{m * n}\right)^z\right) * f + q * p_1 \quad (IV.25)$$

Beispiel IV.2:

Für $m = n = 1000$, $k = l = 10$, $z = 6932$, $q = 0.05$ ($f=1-q=0.95$) und $p_1 = 0.33$ ist $P_{on,f} = 0.4915$.

Es lassen sich analog zu Abschnitt IV.6 die Fehler a) und b), sowie $E(N_t)$, $E(N_A)$ und $E(I)$ herleiten. Für $p_1 = 0$ und $q * 0$ erhält man die Formeln IV.19 - IV. 24. Ist $q = 0$, so ergeben sich die Formeln für die Binomial-Verteilung aus Abschnitt IV.5. Für den Fehlertyp a) gilt:

$$P(a \geq TH_j, O_j^t = 0) = \sum_{i=0}^{TH_j-1} \binom{TH_j-1}{i} * (P_{on,f})^i * (1-P_{on,f})^{1-i} \quad (IV.26)$$

und für den Typ b):

$$P(a \geq TH_j, O_j^t = 0) = \sum_{i=0}^{TH_j-1} \binom{TH_j-1}{i} * (q + p_1 * q)^i * (1 - q - q * p_1)^{1-i}. \quad (IV.27)$$

Setzt man Gleichung IV.26 und Gleichung IV.27 in Gleichung IV.7 ein, so gilt:

$$E(N_t)_f = (m-k) * \sum_{i=TH_j}^1 \binom{1}{i} * (p_{on,f})^i * (1-p_{on,f})^{1-i} + k * \sum_{i=0}^{TH_j-1} \binom{1}{i} * (\delta + q * p_1)^i * (1 - \delta - q * p_1)^{1-i}. \quad (IV.28)$$

Minimierung von $E(N_t)_f$ bezüglich TH führt zu der **unteren Grenze** des gesamten Fehlers $E(N_A)_f$ bei gestörten Verbindungen:

$$E(N_A)_f = \min(E(N_t)_f) * z. \quad (IV.29)$$

Für die Werte aus Beispiel IV.2 ergeben sich

$$P(a \geq 10, O_j^t = 0) = 8.234 * 10^{-4}, P(a \geq 9, O_j^t = 0) = 9.341 * 10^{-3}$$

$$P(a < 10, O_j^t = 1) = 0.2888, P(a < 9, O_j^t = 1) = 0.0422, \min(E(N_t)_f) = 3.70$$

und somit $E(N_A)_f = 25670$. In den Ausgabemustern werden dabei im Mittel 0.81 zusätzliche Einsen erzeugt und 2.9 Einsen weggelassen.

Die Informationsspeicherkapazität $E(I)_f$ ist abschätzbar durch Einsetzen von $\min(E(N_t)_f)$ in Gleichung IV.16 durch:

$$E(I)_f \leq -z \sum_{i=0}^{k-1} \text{Id} \left(\frac{\min(E(N_t)_f) + k - i}{m - i} \right). \quad (IV.30)$$

IV.8.1 Zuverlässigkeits Kenngröße eines Assoziativspeichers mit gestörten Verbindungselementen

Um die Zuverlässigkeit eines Assoziativspeichers einschätzen zu können, benötigt man Kenngrößen, die als Maßstab für die Zuverlässigkeit dienen. Diese Kenngrößen werden im folgenden eingeführt. Die erste Kenngröße, die 'lokale Assoziationszuverlässigkeit' (LAZ) für gestörte Verbindungselemente wird definiert durch:

$$\omega_b := \frac{1}{(1+f) * E(N_t)_f + 1} = \frac{1}{(2-b) * E(N_t)_f + 1}. \quad (IV.31)$$

Ist eine lokale Assoziationszuverlässigkeit ω_{\min} vorgegeben, die erreichbar ist, so läßt sich mit Hilfe eines numerisches Verfahrens ein z berechnen, so daß $\omega_{\min} < \omega_b$ ist. Für $E(N_t)_f < 1$ aus Abschnitt IV.5.2 erhält man:

$$\omega_{\min} = \frac{1}{1 + (1+f)} < \omega_{\delta}. \quad (\text{IV.32})$$

Desweiteren wird die 'globale Assoziationszuverlässigkeit' (GAZ) für gestörte Verbindungselemente definiert durch:

$$\Omega_b := \frac{1}{(1+f) * E(N_A)_{f+1}} = \frac{1}{(2-b) * E(N_A)_{f+1}}. \quad (\text{IV.33})$$

Durch Vorgabe eines Ω_{\min} (s.o.) erhält man ebenfalls mit Hilfe eines numerischen Verfahrens ein z mit $\hat{\Omega}_{\delta} > \Omega_{\min}$. Für den Ansatz aus Abschnitt IV.5.1 ($E(N_A) < 1$) gilt:

$$\Omega_{\min} = \frac{1}{1 + (1+f)} < \Omega_{\delta}. \quad (\text{IV.34})$$

Abbildung IV.10 zeigt den Verlauf der lokalen und globalen Assoziationszuverlässigkeit in Abhängigkeit vom Grad der Verbindungen f für $m = n = 500$, $l = k = 14$ und $z = 600$. Die globale Assoziationszuverlässigkeit liegt weit unterhalb der Gerade, die durch Gleichung IV.34 gegeben ist, d.h. die Zuverlässigkeit kann nicht erreicht werden, da die Muster-

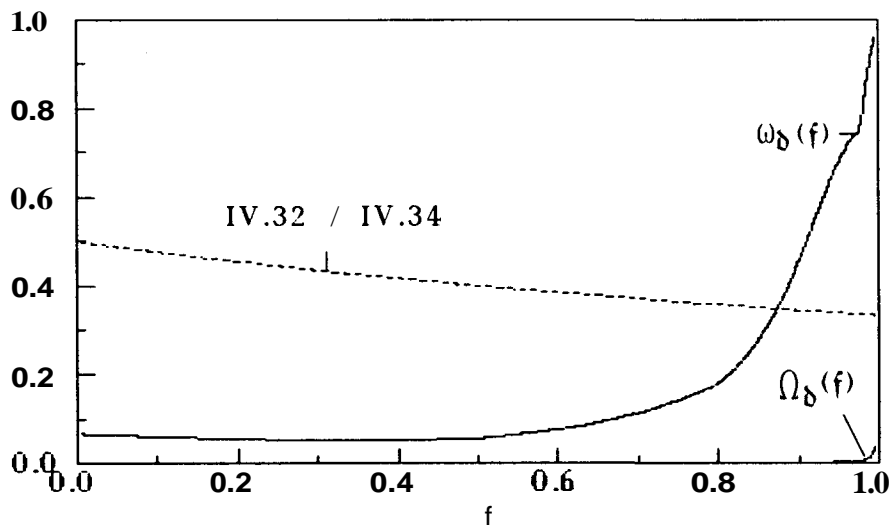


Abbildung IV.10: Lokale und globale Assoziationszuverlässigkeit ω_{δ} , Ω_{δ} in Abhängigkeit vom Grad der Verbindung f ($m = n = 500$, $l = k = 14$, $z = 600$ und $p_1 = 0.8$).

anzahl zu hoch gewählt wurde. Die lokale Zuverlässigkeit kann für große f erfüllt werden, fällt dann aber mit Anzahl der Verbindungen unter die geforderte Grenze. Abbildung IV.11 zeigt die Kurven für $z = 300$. Für große f läßt sich auch die globale Assoziationszuverlässigkeit erfüllen. Der Speicher ist fehlertoleranter, da die Assoziationszuverlässigkeit für größere b noch erfüllt ist.

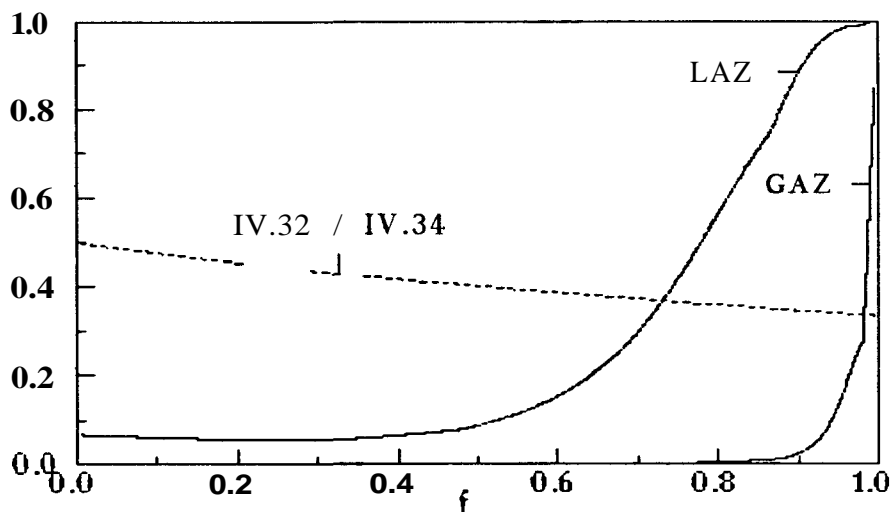


Abbildung IV.11: Lokale und globale Assoziationszuverlässigkeit im Abhängigkeit vom Grad der Verbindung f . $m = n = 500$, $l = k = 14$, $z = 300$ und $p_1 = 0.5$.

Die Assoziationszuverlässigkeit ist eine sehr hilfreiche Kenngröße beim Testen von Assoziativspeichern. Will man wissen, ob der Assoziativspeicher mehr als f Verbindungen enthält, ohne ihn vollständig zu testen, dann erhält man durch Abbildungen der Art IV.10 und IV.11 eine Assoziationszuverlässigkeit für eine gewisse Anzahl z von Mustern. Speichert man die z zufällig erzeugten Muster in den Assoziativspeicher und erhält eine Bitfehleranzahl unterhalb der durch die Assoziationszuverlässigkeit vorgegebenen Bitfehleranzahl, dann enthält der Speicher wahrscheinlich mehr als f Verbindungen.

Beispiel IV.3:

Will man einen Speicher mit $m = n = 500$ Verbindungen testen, so wählt man für einen Test 300 zufällige Musterpaare der Länge 500 mit jeweils 14 Einsen. Der Speicher sollte mehr als 220000 korrekt arbeitende Verbindungselemente (von 250000) enthalten, d.h. $f = 0.88$. Man liest aus Abbildung IV.11 eine lokale Assoziationszuverlässigkeit von 0.8 ab. Dies bedeutet, daß während des Auslesens der 300 Muster nicht mehr als 39 Bitfehler auftreten dürfen.

IV.9 Simulationsergebnisse

IV.9.1 Simulation eines Assoziativspeichers mit korrekt arbeitenden Verbindungselementen

Um die theoretischen Ableitungen beurteilen und verifizieren zu können, wurde ein Simulationswerkzeug entwickelt, mit dem Assoziativspei-

cher mit bis zu 5000 x 5000 Verbindungen simuliert werden können. Zur Unterstützung der Auswertung von Simulationsergebnissen wurden im Rahmen dieser Arbeit weiterhin umfangreiche Grafikroutinen implementiert, die es erlauben, die Simulationsergebnisse angemessen darzustellen. Das Simulationskonzept wird in Kapitel VII ausführlich beschrieben.

Ein Problem bei Simulationen mit Hilfe von Rechnern ist die Erzeugung von Zufallszahlen. Für diese Simulationen wurde ein *Fibonacci-Generator* implementiert. Die Zufallszahlen r_n werden generiert durch:

$$r_n = X_n / M; \quad X_{n+1} = (X_n + X_{n-1}) \bmod M, \text{ mit } M = 2^{23}. \quad (\text{IV.35})$$

und sind im Intervall [0..1] gleichverteilt (ZIE78).

Als erstes wurde ein einfacher Assoziativspeicher simuliert mit $m = n = 100$, $l = k = 10$, $b = 0$, d.h. alle Verbindungen sind möglich. Es wurden $z_{\max} = 200$ Musterpaare zufällig erzeugt (Hetero-Assoziation). Durchgeführt wurden jeweils $n = 100$ Simulationsläufe, wobei für jedes $z \in \{1..z_{\max}\}$ $E(I)$, $E(N_A)_{\min}$, p_{ON} und der Schwellenwert $TH_j = TH$, für $j \in \{1..m\}$ bei minimalem $E(N_t)$ gemessen wurden. Als Ergebniss wurde ein Punktschätzwert für den Mittelwert $E(X)$ und die Standardabweichung $D(X)$ für $E(I)$, $E(A)_{\min}$, p_{ON} und TH , $z \in [1..z_{\max}]$ berechnet durch:

$$\bar{x} = E(X) = \frac{1}{n} * \sum_{i=1}^n x_i \text{ und} \quad (\text{IV.36})$$

$$\beta^2 = D(X) = \frac{1}{n-1} * \sum_{i=1}^n (x_i - \bar{x})^2; \quad (\text{IV.37})$$

Die Abbildungen IV.12 - IV.14 zeigen die Ergebnisse der ersten Simulation. In Abbildung IV.12 sind die gemessenen Mittelwerte $p_{ON}(z)$ und die nach Gleichung IV.1 berechneten Werte $p_{ON,IV.1}(z)$ aufgetragen, ebenfalls die Standardabweichung vom Mittelwert nach oben und unten. Die Approximation der Simulationswerte durch Gleichung IV.1 ist so gut, daß alle Werte in der Abbildung IV.12 auf einer Linie erscheinen.

Abbildung IV.13 zeigt die gemessenen Mittelwerte und die durch die Gleichung IV.10 und IV.11 berechneten Werte für $E(N_A)(z)$. Es zeigt sich, daß die von Palm berechnete Bitfehlerwahrscheinlichkeit zu etwas

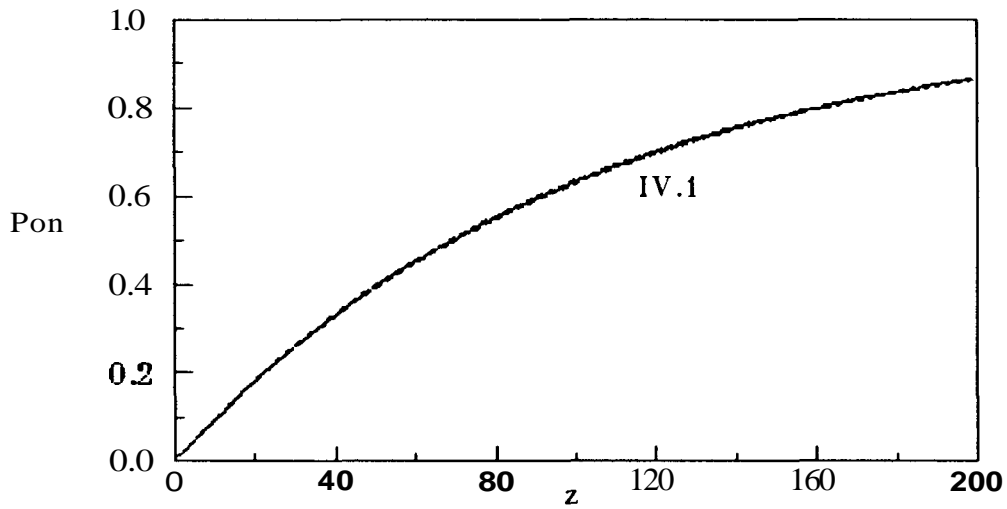


Abbildung IV.12: Belegungswahrscheinlichkeit p_{0n} in Abhängigkeit von der Anzahl der gespeicherten Muster z .

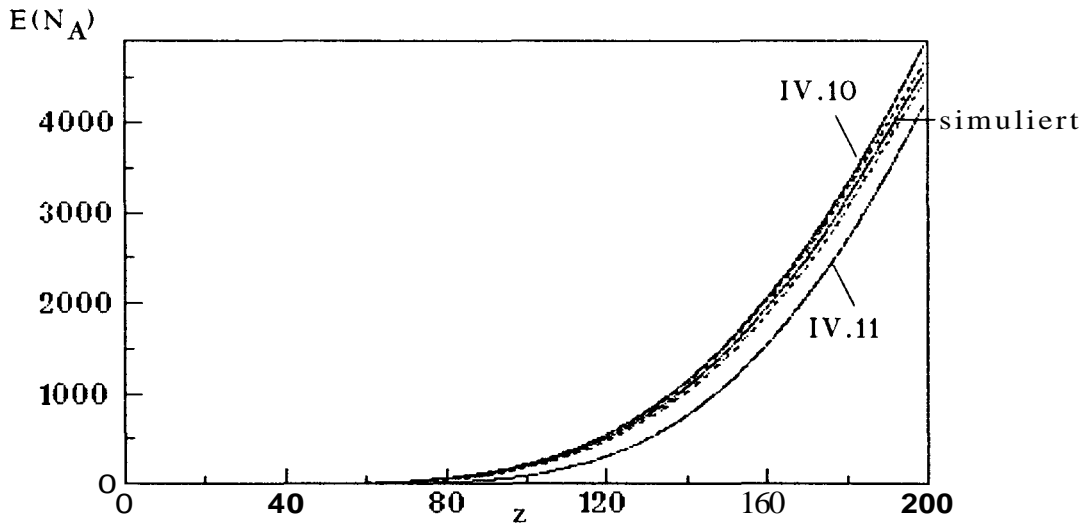


Abbildung IV.13: Gesamter Bitfehler $E(N_A)$ in Abhängigkeit von der Anzahl der gespeicherten Muster z .

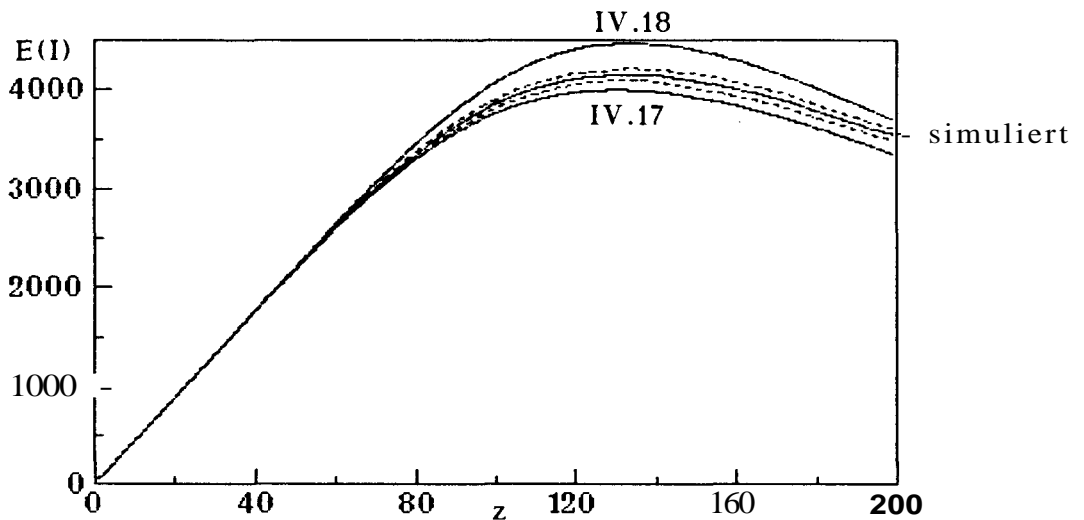


Abbildung IV.14: Informationsspeicherkapazität $E(I)$ in Abhängigkeit von der Anzahl der gespeicherten Muster z .

höheren Gesamtfehlern $E(N_A)$ führt, während die dieser Arbeit angenommene Binomial-Verteilung zu niedrigeren Gesamtfehlern $E(N_A)$ führt, die allerdings weiter vom gemessenen Mittelwert entfernt liegen. Der Grund liegt in der niedrigen Anzahl von Verbindungen (100×100), für die $p_{0n} \neq p$ ist. Erst asymptotisch geht p_{0n} gegen p . Die gestrichelten Linien zeigen die Standardabweichung vom Mittelwert.

Abbildung IV.14 zeigt abschließend die Informationsspeicherkapazität $E(I)(z)$. Hier zeigt sich, daß die Bitfehlerwahrscheinlichkeit $p(z)$ von Palm zu einer geringeren Informationsspeicherkapazität als die gemessenen Mittelwerte $E(I)(z)$ führt. Dagegen berechnet sich mit der Binomial-Verteilung (p_{0n}^1) eine höhere Informationsspeicherkapazität, was aufgrund der Verteilung der Bitfehler (Abbildung IV.13) zu erwarten ist.

Tabelle IV.3 zeigt für die Musteranzahlen z_1, \dots, z_5 aus den Gleichungen IV.10 - IV.14 die Informationsspeicherkapazität $E(I)$, die Belegungswahrscheinlichkeit p_{0n} und die Gesamtanzahl der Fehler $E(N_A)$. Es ergibt sich eine relativ große Abweichung der Musteranzahl bezüglich der Gleichungen IV.10 und IV.11 (29%), weil die Binomial-Verteilung nur für große Assoziativspeicher eine gute Approximation darstellt. Für $m = n = 1000$, und $k = 1 = 4$ beträgt die Abweichung nur noch 10.9% (siehe Tabelle IV.2). Entsprechendes gilt für die Gleichungen IV.8. Auf die Informationsspeicherkapazität hat dieser Fehler offensichtlich keinen großen Einfluß. Sehr gut wird p_{0n} durch Gleichung IV.1 berechnet.

Für die Simulationen mit gestörten und zerstörten Verbindungselementen ist zu erwarten, daß die durch Gleichung IV.23 berechnete Fehleranzahl unterhalb der durch die Simulation berechneten Fehleranzahl bleibt.

	z_1	z_2	z_3	z_4	z_5
z	39	55	83	99	69
$E(N_A)_1$	0.8972	7.2735	78.9	205.25	27.6536
$E(N_A)_2$	0.0451	0.9438	25.1021	88.13	6.0842
$E(N_A)_{sim}$	0.82	6,92	74.24	189.98	25.14
$E(I)_1$	1711.3	2389	3375	3738,6	2927.9
$E(I)_2$	1714.9	2414.7	3551.4	4043.4	3009.3
$E(I)_{sim}$	1712.27	2395.45	3423.63	3832.31	2952.23
p_{0n}	0.3243	0.4246	0.5658	0.6303	0.50
$p_{0n,sim}$	0.3237	0.4245	0.5653	0.6294	0.5001

Tabelle IV.3: Ausgewählte Werte der Informationsspeicherkapazität, gesamten Bitfehleranzahl und der Belegungswahrscheinlichkeit. Mit sim sind die bei der Simulation berechneten Werte gekennzeichnet ($m = n = 100$, $k = 1 = 10$).

IV.9.2 Simulation eines Assoziativspeichers mit zufällig zerstörten Verbindungselementen

Das zweite Simulationsexperiment simuliert einen Assoziativspeicher mit $m = n = 100$, $k = l = 10$, $b = 0.1$, d.h. $\delta = 0.9$ und $p_1 = 0$. Analog zu Abschnitt IV.9.1 wurden wieder Punktschätzer für Mittelwert und Standardabweichung ermittelt und in Abbildung IV.15 bis IV.17 für $p_{on,\delta}$, $E(A)_\delta$ und $E(I)_\delta$ dargestellt. Die gestrichelten Linien stellen die Standardabweichungen nach oben und unten des durch die Simulation gewonnenen Mittelwertes dar. Die andere Linie zeigt die durch die Formeln berechneten Werte. Erwartungsgemäß weicht die durch die Simulation gewonnene Belegungswahrscheinlichkeit kaum von der durch die Formeln berechneten ab.

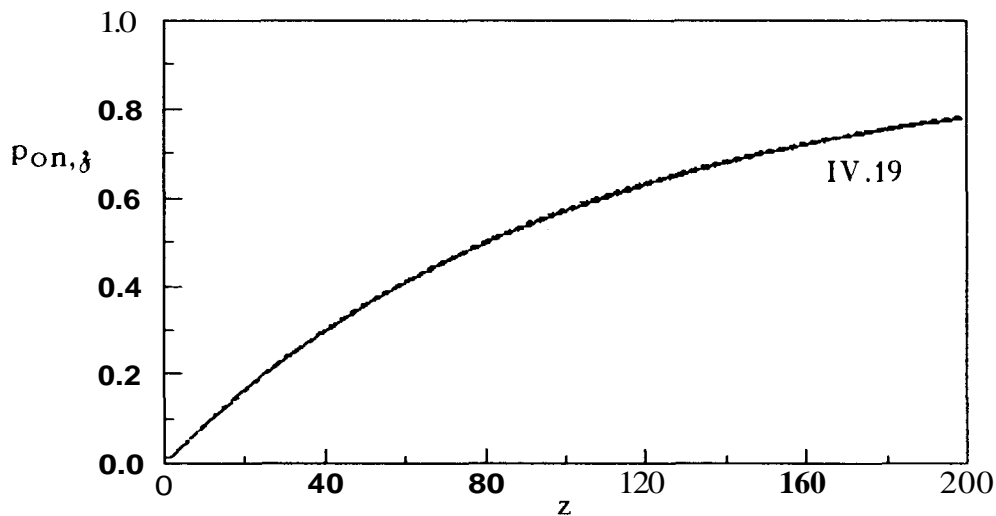


Abbildung IV.15: Belegungswahrscheinlichkeit $p_{on,\delta}$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 10% zerstörten Verbindungselementen.

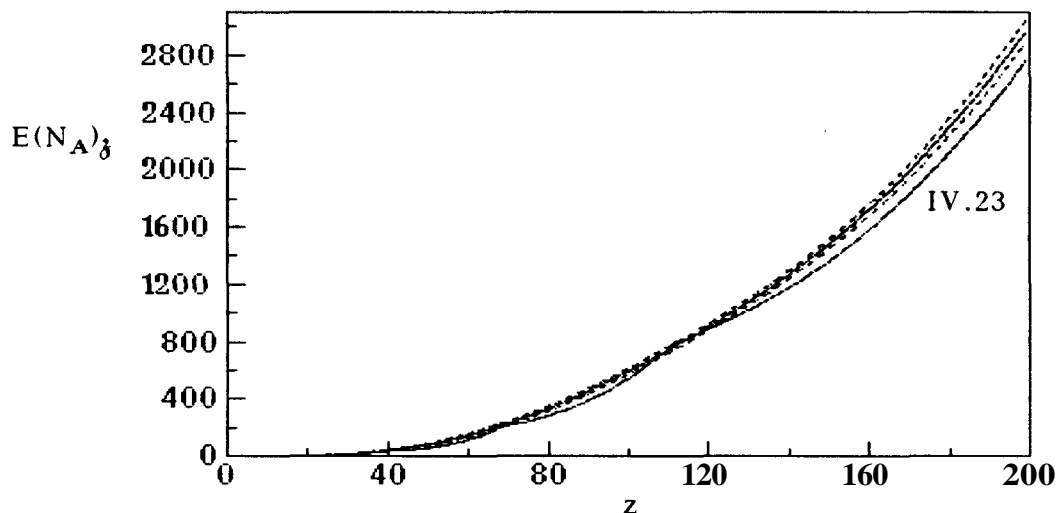


Abbildung IV.16: Gesamter Bitfehler $E(N_A)_\delta$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 10% zerstörten Verbindungselementen.

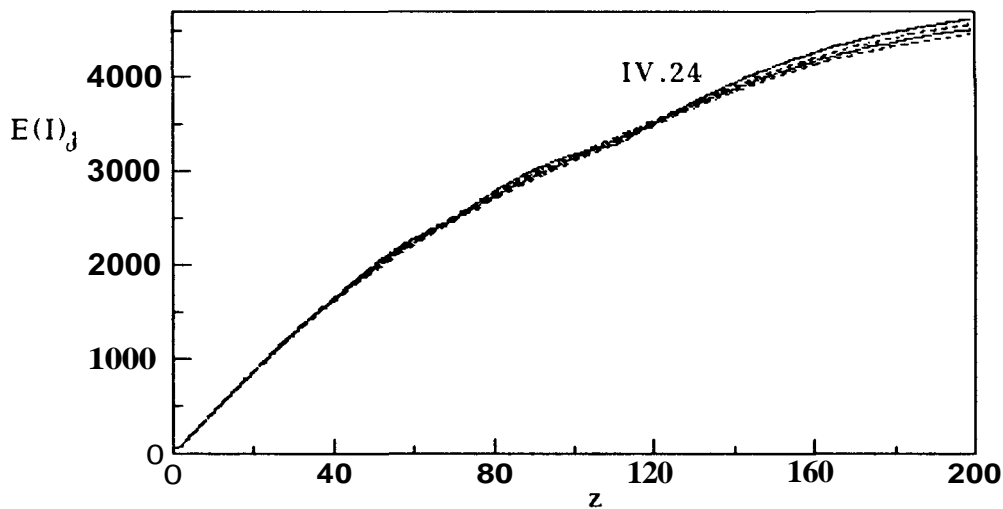


Abbildung IV.17: Informationsspeicherkapazität $E(I)_j$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 10% zerstörten Verbindungselementen.

Die Informationsspeicherkapazität $E(I)_j$ in Abbildung IV.17 ist für $z > 60$ nicht besonders aussagekräftig, denn der Assoziativspeicher berechnet das Ausgabemuster \mathbf{O}^* für große z immer mit der Schwelle $TH = 1 = 10$. Dies bedeutet, daß im Ausgabemuster eine konstante Anzahl zusätzlicher Einsen erzeugt wird, da $P(a > 1, O_j^* = 0)$ gegen $p_{0n}^* \cdot j'$ geht. Gleichzeitig wird auch eine konstante Anzahl von zusätzlichen Nullen erzeugt, da $P(a < 1, O_j^* = 1)$ gegen $1 - j'$ geht. Die erzeugten Ausgabemuster enthalten eine konstante Anzahl von Bitfehlern $E(N_t)$ und die Informationsspeicherkapazität steigt weiter, falls $\text{Id}\binom{m}{k} > \text{Id}\binom{E(N_t)+k}{k}$ ist.

Tabelle IV.4 zeigt die berechneten und die simulierten Werte von $E(N_A)$, $E(I)$ und p_{0n} für die Musteranzahl z_1 aus der numerisch gelösten Ungleichung $1 > E(N_A)$ und z_2 aus der Ungleichung $1 > E(N_t)$. Die Bitfehlerwahrscheinlichkeit wurde mit Hilfe der Binomial-Verteilung berechnet. Die berechneten Werte z_1 und z_2 liegen beide unterhalb der durch die Simulation gewonnenen Werte.

	z_1	z_2
V	15	45
$E(N_A)$	0.9631	43.7139
$E(N_A)_{\text{sim}}$	1.14	66.12
$E(U)$	655.6	1827
$E(I)_{\text{sim}}$	655.9	1801.4
p_{0n}	0.1259	0.3274
$p_{0n, \text{sim}}$	0.1266	0.329

Tabelle IV.4: Ausgewählte Werte der Informationsspeicherkapazität, Gesamtfehleranzahl und Belegungswahrscheinlichkeit. Mit sim sind die bei der Simulation berechneten Werte gekennzeichnet. $m = n = 100$, $k = l = 10$, $b = 0.1$, $p_1 = 0$.

IV.9.3 Simulation eines Assoziativspeichers mit zufällig gestörten Verbindungselementen

Das dritte, vierte und fünfte Simulationsexperiment simuliert einen Assoziativspeicher mit $m = n = 100$, $k = l = 10$, $\delta_1 = 0.01$, $\delta_2 = 0.05$, $\delta_3 = 0.1$ und $p_1 = 0.5$, d.h. es wurde angenommen, daß, wenn ein Verbindungselement ausfällt, es mit der gleichen Wahrscheinlichkeit auf Null oder Eins ausfällt. Die Abbildungen IV.18 - IV.20 zeigen die Ergebnisse der drei Simulationsläufe, wobei jeweils 100 Simulationsläufe für $z_{\max} = 100$ Muster durchgeführt wurden. Die gestrichelten Linien stellen wieder die Standardabweichung um dem Mittelwert dar. Die durch die Formeln berechneten Werte für $E(N_A)_f$ liegen unterhalb der Mittelwerte und die für $E(I)_f$ berechneten Werte oberhalb der Mittelwerte.

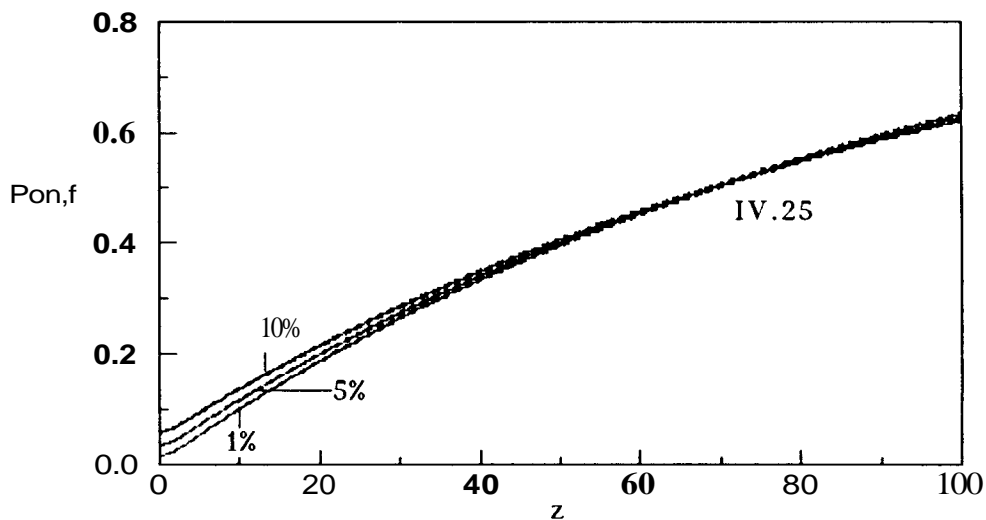


Abbildung IV.18: Belegungswahrscheinlichkeit $p_{On,f}$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 1%, 5% und 10% gestörten Verbindungselementen.

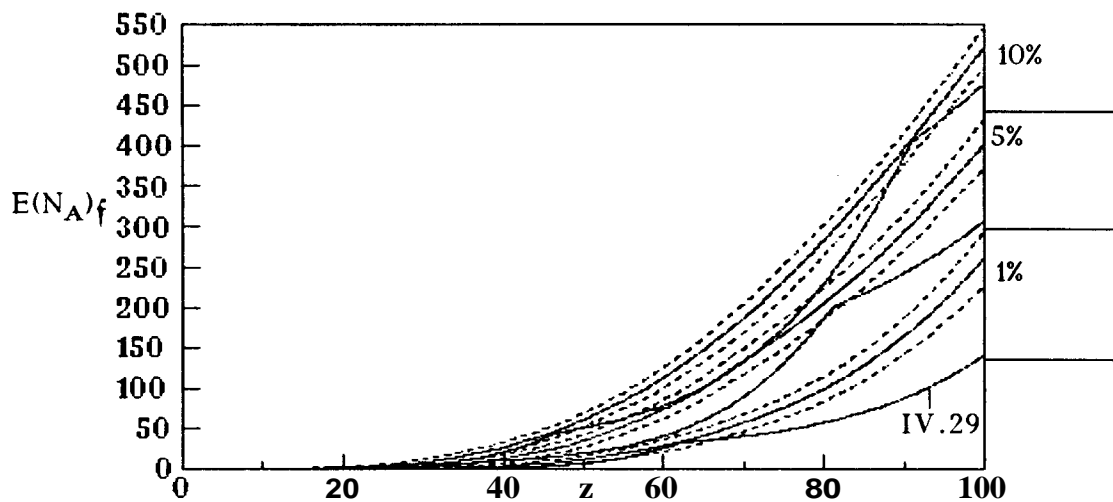


Abbildung IV.19: Gesamte Bitfehler $E(A)_f$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 1%, 5% und 10% gestörten Verbindungselementen.

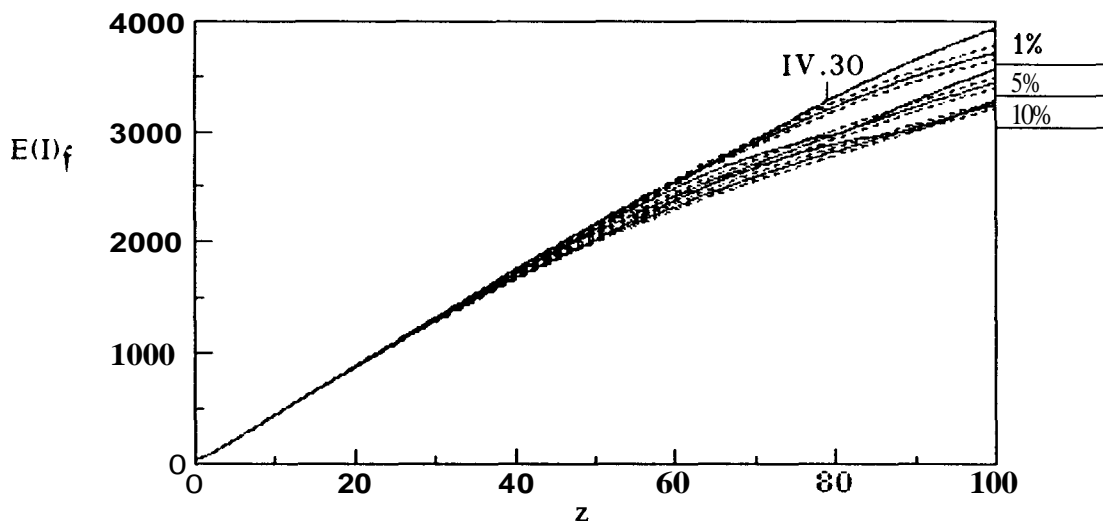


Abbildung IV.20: Informationsspeicherkapazität $E(I)_f$ in Abhängigkeit von der Musteranzahl z eines Assoziativspeichers mit 1%, 5% und 10% gestörten Verbindungselementen.

Wie zu erwarten, zeigen die Simulationsläufe, daß, je mehr Verbindungen gestört sind, die Informationsspeicherkapazität $E(I)_f$ fällt und die gesamte Anzahl der Fehler $E(N_A)$ steigt. Für die Anzahl der Muster $z < 60$ stimmen die berechneten Werte gut mit den simulierten Werten überein, d.h. sie liegen noch innerhalb der Standardabweichung. Erst für $z > 80$ werden die simulierten Werte schlecht approximiert. In diesem Fall treten aber schon so viele Fehler auf, daß er für die Praxis uninteressant ist. Die simulierte Belegungswahrscheinlichkeit $p_{on, sim}$ wird wiederum am besten durch die berechneten Werte approximiert.

Abbildung IV.21 zeigt, wie die Speicherkapazität eines Assoziativspeichers in Abhängigkeit von der Zerstörung der Verbindungen fällt. Die Kurve a) wurde durch jeweils 50 Simulationsläufe bei jedem $q = 0, 0.01, 0.02, \dots, 0.2$ gewonnen. Es zeigt sich, daß schon bei einer Störung von 10% weniger als die Hälfte der Muster fehlerfrei gespeichert werden kann.

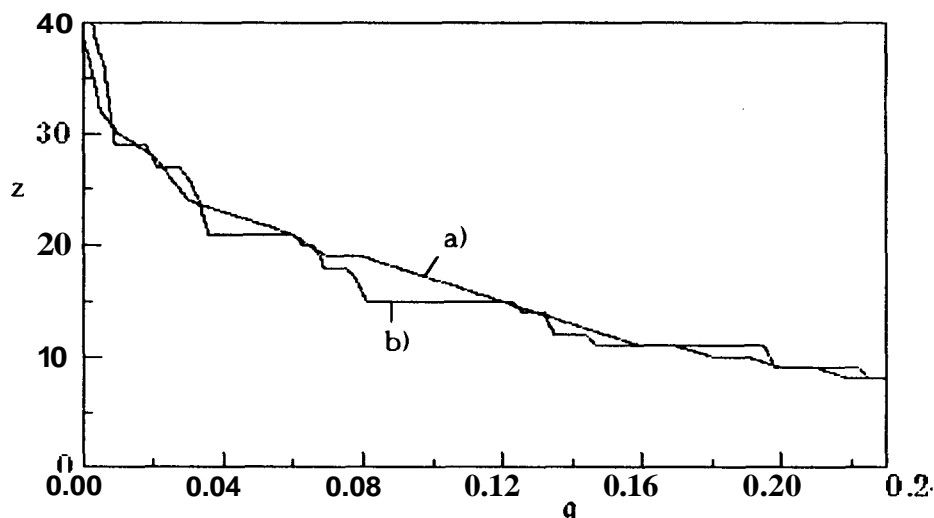


Abbildung IV.21: Anzahl der fehlerfrei einspeicherbaren Muster z in Abhängigkeit von der Zerstörung p für $m = n = 100, k = 1 = 10, p_1 = 0.5$. a) Mittelwert der Simulationen, b) berechnete Kurve.

IV.10 Ableitungen für gestörte Eingabevektoren

IV.10.1 Ausfall einer Eins im Eingabemuster

Der Assoziativspeicher kann auch für ein gestörtes Eingabemuster das korrespondierende Ausgabemuster berechnen (siehe Abbildung IV.5). Eine Störung liegt z.B. vor, wenn das Eingabemuster \underline{P} an der Position i eine Null enthält, obwohl im Eingabemuster \underline{I}^t an der Position i eine Eins steht, d.h.

c) $I_j^t = I_j^t$ und $I_i^t = 1, I_i^{t'} = 0, \forall j = 1..i-1, i+1,..n$ und einem $i \in \{1..n\}$ beliebig.

Störungen dieser Art sind für den **Auto-Assoziationsfall** ausführlich untersucht und simuliert worden von A. Lansner und Ö.Ekeberg (LAE85). Die von ihnen angestellten Überlegungen werden im folgenden mit Hilfe der vorher abgeleiteten Gleichungen dargestellt und für den **Hetero-Assoziationsfall** verallgemeinert.

Es sei l wieder die Anzahl der Einsen im Vektor \underline{I}^t und l_0 die Anzahl der Einsen im Vektor \underline{I}^t mit $l_0 < l$. Das Ausgabemuster kann nur korrekt ausgelesen werden mit der Schwelle $TH = l_0$, wobei man die Wahrscheinlichkeit für einen Bitfehler vom Typ a) beim Auslesen durch Einsetzen von l_0 in Gleichung IV.3 erhält mit:

$$P(a \geq l_0, O_j^t = 0) = p_{l_0} = 1 + \sum_{h=1}^{l_0} (-1)^h \binom{l_0}{h} \left(1 - \frac{h}{n}(1 - T(m, l, h))\right)^{z-1} \quad (IV.38)$$

Im Auto-Assoziationsfall, d.h. $m = n$ und $l = k$, vereinfacht sich Gleichung IV.38 entsprechend.

Die Wahrscheinlichkeit für einen Bitfehler des Typs b) $P(a < l_0, O_j^t = 1)$ ist wieder gleich Null, denn alle Verbindungen sollen fehlerfrei arbeiten, d.h. $\beta = 1$ (analog zu Abschnitt IV.5).

Ist die Aktivität a_j eine binomial verteilte Zufallsvariable, so gilt asymptotisch:

$$P(a \geq l_0, O_j^t = 0) = \sum_{i=l_0}^{l_0} \binom{l_0}{i} * P_{0n} * (1 - P_{0n})^{l_0-i} = P_{0n} \quad (IV.39)$$

Es läßt sich genau wie in Abschnitt IV.5 mit Hilfe von Gleichung IV.39 die Anzahl der Bitfehler $E(N_t)$ berechnen zu:

$$E(N_t)_{l_0} = (m - k) * p_{l_0} \approx (m - k) * P_{0n} \quad (IV.40)$$

Lansner und Ekeberg definieren nun die 'local recall reliability' (LRR) durch:

$$LLR = \prod_{l_0} = \frac{1 - l_0}{E(N_t)_{l_0} + 1 - l_0} \quad (IV.41)$$

und die 'global recall reliability' als Produkt der lokalen Zuverlässigkeit:

$$GGR = \prod_{i=0}^{1-1} \prod_i \quad (IV.42)$$

Der Wert \prod_{l_0} kann als Zuverlässigkeitskenngröße für die Mustervervollständigung angesehen werden, bei der nur l_0 Einsen des Musters bekannt sind. Ist eine lokale Zuverlässigkeit \prod_{\min} vorgegeben, so erhält man eine analytische Beziehung zwischen der Anzahl der Muster z und der Zuverlässigkeit durch:

$$\prod_{\min} = \frac{1 - l_0}{(m - k) * P_{\dots}^{l_0} + 1 - l_0} \quad (IV.43)$$

$$\Leftrightarrow z \leq \frac{\ln \left(1 - \left(\prod_{\min} \right)^{\frac{1 - l_0}{m - k}} \right)^{1/l_0}}{\ln \left(1 - \frac{l_0 * k}{m * n} \right)} \quad (IV.44)$$

Lansner und Ekeberg erhalten für den Auto-Assoziationsfall ($m = n$, $k = 1$) und der stärksten Einschränkung $E(N_t)_{l_0} \leq n * p_{on}^{l_0}$ (LAE85):

$$z \leq \frac{\ln \left(1 - \left(\prod_{\min} - 1 \right) \frac{1 - l_0}{n} \right)^{1/l_0}}{\ln \left(1 - \left(\frac{1}{n} \right)^2 \right)} \quad (IV.45)$$

Bessere Ergebnisse erhält man durch die Verwendung von p_l anstatt $p_{on}^{l_0}$ und einer numerischen Lösung von Gleichung IV.43, da $p_{on}^{l_0}$ nur asymptotisch gegen p_l geht (siehe Abbildung IV.22). Fordert man, daß möglichst kein Fehler im gesamten Speicher auftreten darf (analog Abschnitt IV.5.1), so ist die lokale Zuverlässigkeit neu zu definieren durch:

$$LLR_n = \prod_{l_0, n} = \frac{1}{E(N_A) + 1 - l_0} \quad (IV.46)$$

Die beiden Gleichungen, die sich durch die Vorgabe eines $\prod_{\min, n}$ (analog Gleichung IV.43) ergeben, lassen sich nur numerisch lösen und ergeben ein wesentliche kleinere Musteranzahl z , da $E(N_A)_{l_0} = z * E(N_t)_{l_0}$.

Die Abbildung IV.22 aus (LAE85) zeigt den Zusammenhang zwischen gestörter Eingabe und lokaler Zuverlässigkeit für $m = n = 500$, $l = k = 14$ und $z = 600$. Je mehr korrekte Einsen in den Eingabevektoren enthalten sind, desto höher ist die Zuverlässigkeit, die erreicht werden kann. Abbildung IV.23 zeigt für diesen Fall den Verlauf der Informationsspeicherkapazität. Da die Musteranzahl relativ klein ist, läßt sich auch mit gestörten Eingabevektoren sehr viel Information aus dem Assoziativspeicher auslesen.

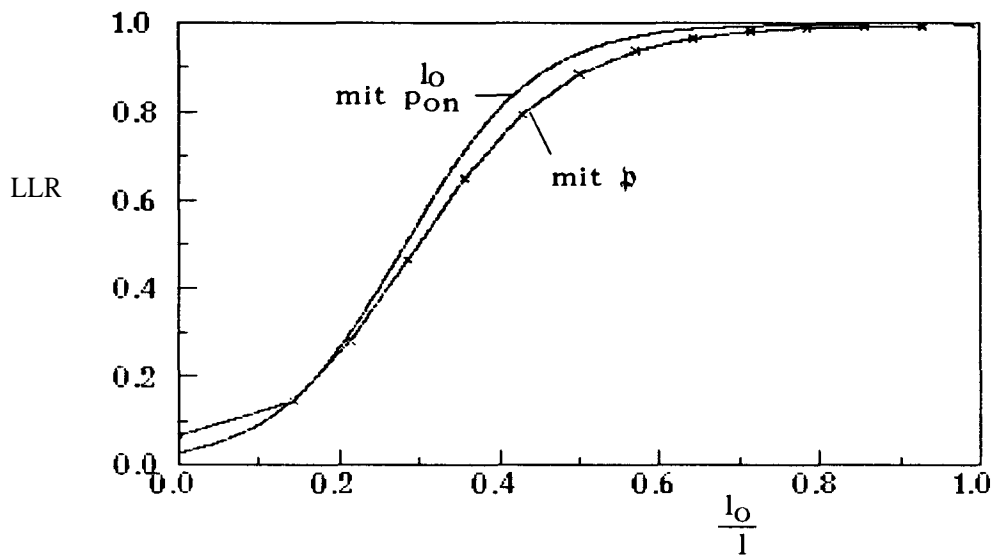


Abbildung IV.22: 'Local recall reliability' in Abhängigkeit der Störung des Eingabevektors nach Lansner und Ekeberg (LAE85) für $m = n = 500$, $k = l = 14$ und $z = 600$ (Auto-Assoziation).

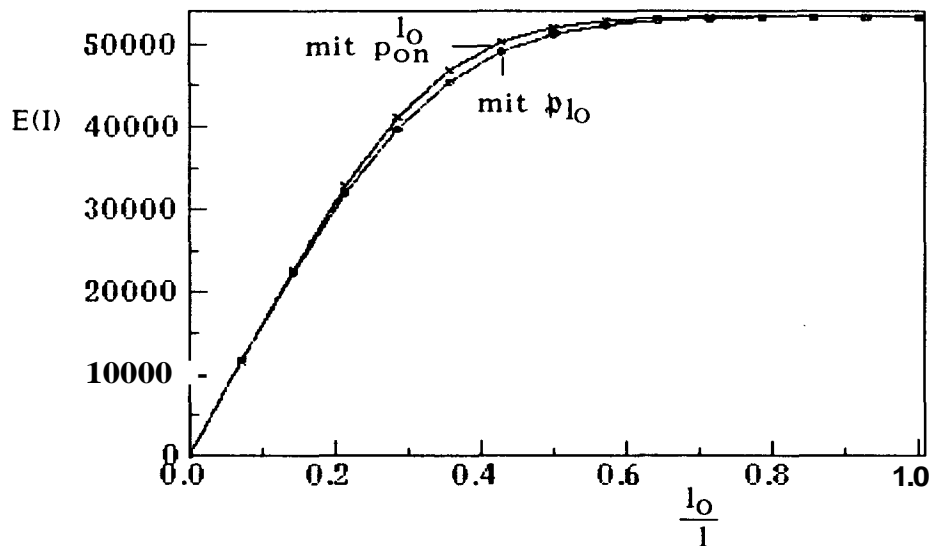


Abbildung IV.23: Informationsspeicherkapazität $E(I)$ in Abhängigkeit der Störung des Eingabevektors für $m = n = 500$, $k = l = 14$ und $z = 600$ (Auto-Assoziation).

IV.10.2 Ausfall einer Eins und zusätzliche Einsen im Eingabemuster

Im Eingabemuster kann nicht nur eine Eins ausfallen, sondern es können auch zusätzliche Einsen erzeugt werden, d.h.

d) $I_j^t = I_j^{t'}$ und $I_i^t = 1, I_i^{t'} = 0$ oder $I_j^t = 0, I_j^{t'} = 1; \forall j \in \{1..i-1, i+1,..n\}$ und ein $i \in \{1..n\}$ beliebig.

Bisher wurde dieser Fall in der Literatur vernachlässigt, in der Technologie ist dieser Fall aber von Bedeutung, denn durch eine Störung können durchaus zusätzliche Einsen erzeugt und nicht nur Einsen ausgeblendet werden. Weiterhin erlangt die Störung noch größere Bedeutung, wenn man rückgekoppelte oder Assoziativspeicher mit mehreren Ebenen betrachtet. Erhält man im rückgekoppelten Fall in der Ausgabe zusätzliche Einsen, so sind dies zusätzliche Einsen der Eingabe im nächsten Rückkopplungsschritt.

Es sei l_0 die Anzahl der Einsen im Vektor \underline{I}^t und l_k die Anzahl der korrekten Einsen im Eingabemuster $\underline{I}^{t'}$. Wird versucht, das Muster \underline{O}^t auszulesen mit einer Schwelle $TH = l' < l_0$, so ergibt sich der Fehler a) $P(A \geq l', O_j^t = 0)$ durch Gleichung IV.38 bzw. Gleichung IV.39, falls eine Binomial-Verteilung vorausgesetzt wird, d.h.:

$$P(A \geq l', O_j^t = 0) = p_{l'} = 1 + \sum_{h=1}^{l'} (-1)^h \binom{l'}{h} \left(1 - \frac{k}{n}(1 - T(m, l_0, h))\right)^{z-1}$$

$$\approx \sum_{i=l'}^l \binom{l'}{i} * p_{0n} * d - p_{0n}^{l-i} \quad (IV.47)$$

und den Fehler b) $P(A < l', O_j^t = 1)$ durch:

$$P(A < l', O_j^t = 1) = \begin{cases} 0, & \text{falls } l' \leq l_k \\ p_1, & \text{sonst} \end{cases} \quad (IV.48)$$

$$\text{mit } p_1 = \sum_{h=1}^{l'-l_k} (-1)^{h+1} \binom{l'-l_k}{h} \left(1 - \frac{k}{n}(1 - T(m, l_0 - l_k, h))\right)^{z-1}$$

$$\approx 1 - \sum_{i=l'-l_k}^{j=l_0-l_k} \binom{j}{i} * p_{0n} * (1 - p_{0n})^{j-i} \quad \text{bei binomial verteilter Aktivität } a_j.$$

Die Anzahl der Fehler pro Muster $E(N_t)$ und die gesamte Anzahl der Fehler $E(N_A)$ ergeben sich durch Einsetzen von Gleichung IV.47 und Gleichung IV.48 in Gleichung IV.8 bzw. Gleichung IV.9. Abbildung IV.24 zeigt den Verlauf der gesamten Anzahl der Fehler $E(N_A)$ in Abhängigkeit von der Musteranzahl z bei verschiedenen Störungen. Im Fall a) hat der Eingabevektor eine korrekte Eins weniger und dafür drei zusätzliche falsche Einsen. Im Fall b) sind es korrekte

Eins weniger und zwei zusätzliche falsche Einsen, während die Kurve c) den ungestörten Fall darstellt. Es zeigt sich, daß, je mehr zusätzliche Einsen erzeugt werden, die gesamte Anzahl der Fehler $E(N_A)$ steigt.

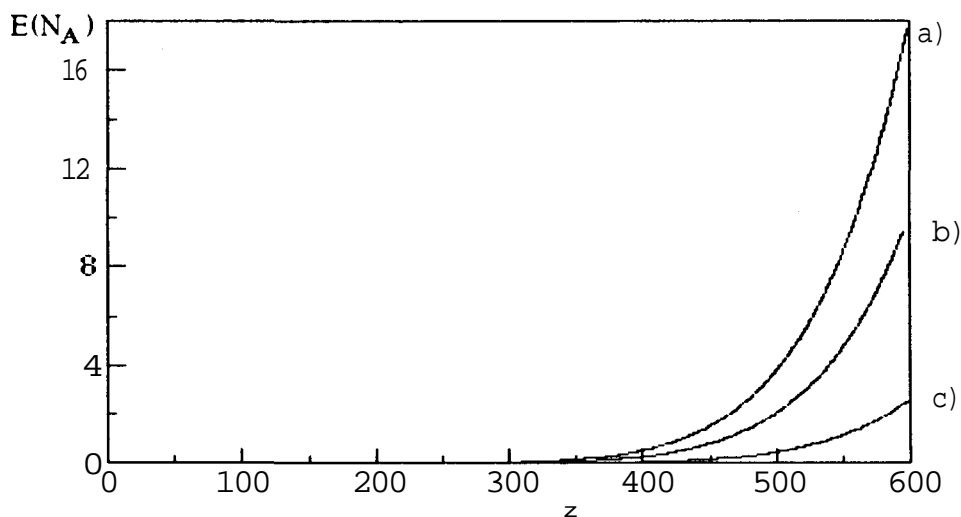


Abbildung IV.24: Gesamt-Fehleranzahl $E(N_A)$ in Abhängigkeit von der Musteranzahl z für $m = n = 500$, $k = l = 14$ und $z = 600$, a) $l_0 = 16$, $l_k = 13$, b) $l_0 = 15$, $l_k = 13$ und c) $l_0 = l_k = 14$.

IV.11 Parametervariation der Verbindungselemente

Werden beim Schaltungsentwurf die Verbindungsgewichte des einfachen Assoziativspeichers analog entworfen, so ist bekannt, daß es aufgrund von herstellungsbedingten Mängeln zu einer Streuung von Parametern in der Fertigung kommt. Im folgenden wird dieser Einfluß auf die Funktionsweise des Assoziativspeichers dargestellt.

Betrachtet man die Ausgabefunktion S des einfachen Assoziativspeichers, so gilt, daß sie eine Eins ergibt, wenn die Aktivität $a = \sum w_{ij} * I_j \geq TH$ ist. Für dieses theoretische Modell wählt man bei binären Gewichten und Eingabevektoren $TH = 1 = \max(a)$, wenn der Eingabevektor l Einsen enthält. Damit ist gewährleistet, daß ein Prozessorelement nur eine Eins ausgibt, wenn alle l Einsen auf aktivierte Verbindungen treffen. In der Praxis ist es aber nicht günstig, $TH = 1$ zu wählen, denn kleine Parameterabweichungen nach unten würden schon dazu führen, daß $l > \max(\sum w_{ij} * I_j)$ und damit keine Eins ausgegeben würde. Günstiger wäre es, $TH \approx 1 - 0.5$ zu wählen, damit der Störabstand s nach oben und unten gleich groß wäre. Im schlimmsten Fall variieren die Gewichte bzw. Eingabegrößen für ein Prozessorelement a , das eine Eins ausgeben soll, nach unten und für ein Prozessorelement b , das noch gerade eine Null ($\sum w_{ib} * I_j = 1 - l$) ausgeben soll, nach oben. Die Parametervariation a soll vom Betrag für alle Elemente gleich sein und tritt nur für aktivierte Elemente auf. Für den Störabstand $s = 1$ gilt dann:

$$s = 1 > \sum w_{ia} * I_j * \sigma + \sum w_{ib} * I_j * \sigma = \sigma * (1 + 1 - 1)$$

<=>

$$\sigma < \frac{s}{2 * 1 - 1} = \frac{1}{2 * 1 - 1}$$

(IV49)

Der Schwellenwert TH muß in diesem Fall folgendermaßen gewählt werden:

$$TH = \sum w_{ia} * I_i - \sum w_{ia} * I_j * (3 = 1 - 1 * \sigma = 1 * (1 - \frac{1}{2 * 1 - 1})),$$

also:

$$TH_{max} = \frac{2 * 1 * (1 - 1)}{2 * 1 - 1}$$

(IV.50)

Tabelle IV.5 zeigt für verschiedene Werte von 1 die maximal zulässige Streuung σ und den zu wählenden Schwellenwert TH in Abhängigkeit von der Anzahl 1 der Einsen im Eingabemuster. Für große 1 wird die maximale Streuung sehr klein, d.h. die Eingabevektoren sollten möglichst spärlich kodiert sein ($O(\log(n))$), damit der Speicher unempfindlicher gegenüber Parametervariation wird. Des Schwellenwert TH geht asymptotisch gegen 1 - 0,5, wobei wichtig ist, daß der Wert sehr genau einstellbar ist, also nicht auch noch variiert.

1	G	TH
2	1/3	4/3
3	1/5	12/5
4	1/7	24/7
5	1/9	40/9
6	1/11	60/11

Tabelle IV.5: Maximal zulässige Streuung σ und Schwellenwert TH in Abhängigkeit von der Anzahl der Einsen 1 im Eingabemuster.

IV.12 Simulation mit ASCII-Mustern

Nach der Untersuchung des einfachen Assoziativspeichers mit zufälligen Mustern, folgt nun die Untersuchung mit praktischen, in der Realität vorkommenden Mustern. Dazu werden 56 ASCII-Muster vorgegeben und zwar die Buchstaben A - Z, die Zahlen 0 - 9 und 20 Sonderzeichen. Die Muster sind als 5 x 7 Punktmatrix abgelegt, die als binärer Vektor der Länge 35 aufgefaßt wird. Im Anhang 1 sind die Muster dargestellt.

IV.12.1 Speicherkapazität eines 35 x 35 Assoziativspeichers

Als erstes wird untersucht, wieviele von den 36 Mustern A - Z und den Zahlen 0 - 9 in einen Assoziativspeicher mit $m = n = 35$ fehlerfrei eingespeichert werden können. Die 36 Muster enthalten im Mittel $l = k = 14.8 \approx 15$ Einsen. Die numerische Lösung von Gleichung IV.10 ergibt eine maximale Musteranzahl von $z \leq 4$. Tabelle IV.6 zeigt die Werte für die Belegungswahrscheinlichkeit, die Informationsspeicherkapazität, die Bitfehleranzahl und die gesamte Fehleranzahl für die Musteranzahlen $z = 4$ und $z = 5$.

V	4	5
p_{0n}	0.556	0.64
$E(I)$	125.4	153.0
$E(N_t)$	0.050	0.218
$E(N_A)$	0.2	1.09

Tabelle IV.6: Berechnete Werte für zufällige Muster bei der Untersuchung eines 35 x 35 Assoziativspeichers mit 36 ASCII-Mustern.

Für zufällige Muster lassen sich im Mittel vier Muster in den Assoziativspeicher fehlerfrei einspeichern. Um die Muster nicht auf sich selbst abzubilden (Auto-Assoziation), werden die Muster jeweils auf den Nachfolger abgebildet, d.h. A -> B, B -> C, 9 -> A (Hetero-Assoziation). Versucht man, die Muster der Reihenfolge nach (A,B..Z,0..9) zu speichern, so lassen sich nur die ersten 3 Muster fehlerfrei speichern. Das vierte Musterpaar wird mit zwei Bitfehlern ausgelesen. Tabelle IV.7 zeigt die gemessenen Werte der Belegungswahrscheinlichkeit p_{0n} , der Informationsspeicherkapazität $E(I)$, der gesamten Fehleranzahl $E(N_A)$, der Anzahl der möglichen Einsen und die Anzahl der Kollisionen, d.h. die Verbindungsgewichtsmatrix war an dieser Stelle schon mit einer Eins belegt.

V	36	ersten 3	ersten 4
Pon	0.93	0.473	0.52
$E(N_A)$	419	0	2
$E(I)$	300.2	94.46	119.0
* Einsen	7889	785	1091
« Kollisionen	6754	206	449

Tabelle IV.7: Werte bei der Untersuchung eines 35 x 35-Assoziativspeichers mit 36 ASCII-Mustern

Um die maximale Anzahl von fehlerfrei speicherbaren Mustern zu finden, werden die Muster gespeichert, die einen großen Hammingabstand (>7) besitzen. Das sind die Muster A, V, K, 0, 1, 4. Sie werden auf ihren Nachfolger abgebildet (A \rightarrow V, V \rightarrow K, ... 4 \rightarrow A, Hetero-Assoziation) und danach eingespeichert. Diese 6 Muster konnten fehlerfrei gespeichert werden, allerdings ließen sich keine 7 Muster finden, die aus dem Assoziativspeicher fehlerfrei ausgelesen werden konnten. Mit dem siebten Muster X ergaben sich die Werte in Tabelle IV.8.

V	ausgesucht 6	ausgesucht 7
Pon	0.604	0.654
$E(N_A)$	0	4
$E(I)$	182.05	199.47
» Einsen	1046	1212
« Kollisionen	306	411

Tabelle IV.8: Werte bei der Untersuchung eines 35 x 35 Assoziativspeichers mit sechs bzw. sieben ausgewählten ASCII-Mustern.

Eine weitere interessante Frage ist, wieviele Muster sich speichern lassen, wenn im Mittel weniger als zwei Bitfehler beim Auslesen in den Ausgabemustern vorkommen sollen, d.h. $E(N_t) < 2$. Für zufällige Muster ist die Lösung in Tabelle IV.9 dargestellt.

V	8	9
Pon	0.80	0.84
$E(I)$	197.5	198.7
$E(N_t)$	1.94	2.93
$E(N_A)$	15.5	26.4

Tabelle IV.9: Berechnete Werte für zufällige Muster bei der Untersuchung eines 35 x 35-Assoziativspeichers mit 36 ASCII-Mustern und der Bedingung $E(N_t) < 2$.

Tabelle IV.10 zeigt die gemessenen Werte für die neun bzw. zehn ausgewählten Muster A, K, V, X, Y, Z, 0, 1, 4, [2]. Abbildung IV.25 zeigt die ausgelesenen Muster.

	ausgesucht 9	ausgesucht 10
Pon	0.76	0.807
$E(N_A)$	13	22
$E(I)$	234.79	240.14
* Einsen	1494	1718
* Kollisionen	564	729

Tabelle IV.10: Werte bei der Untersuchung eines 35 x 35 Assoziativspeichers mit neun bzw. zehn ausgewählten ASCII-Mustern.

IV.12.2 Speicherkapazität bei Kodierung der Eingabevektoren

Ein Ziel ist es möglichst die 36 Muster (A..Z und 0..9) in einen einfachen Assoziativspeicher fehlerfrei zu speichern. Dazu muß die Anzahl der Verbindungen erhöht werden. Es gibt einen relativ einfachen Algorithmus, mit der gewöhnliche Kodierungen aus gleichwertigen Nullen und Einsen in eine ungleiche Folge mit spärlichen Einsen umkodiert werden können. (PAL88) Dazu unterteilt man den Eingabevektor in j -Abschnitte der Länge x . Jeden dieser j -Abschnitte faßt man als Binärzahl auf, also als eine Zahl zwischen 0 und 2^x . Die Zahl kodiert man als Folge der Länge 2^x , die nur eine einzige Eins an der Stelle enthält, die dem Wert der Zahl entspricht.

Beispiel IV.4:

$x = 3$, 1-ter Abschnitt = 1102 als Binärzahl entspricht einer 6_{10} im Zehnersystem und daraus ergibt sich die kodierte Folge 00100000.

Durch das Verfahren erhält man eine wesentlich längere Folge von Nullen und Einsen (anstatt $j \cdot x$ enthält die Folge jetzt $j \cdot 2^x$ Glieder) mit genau j

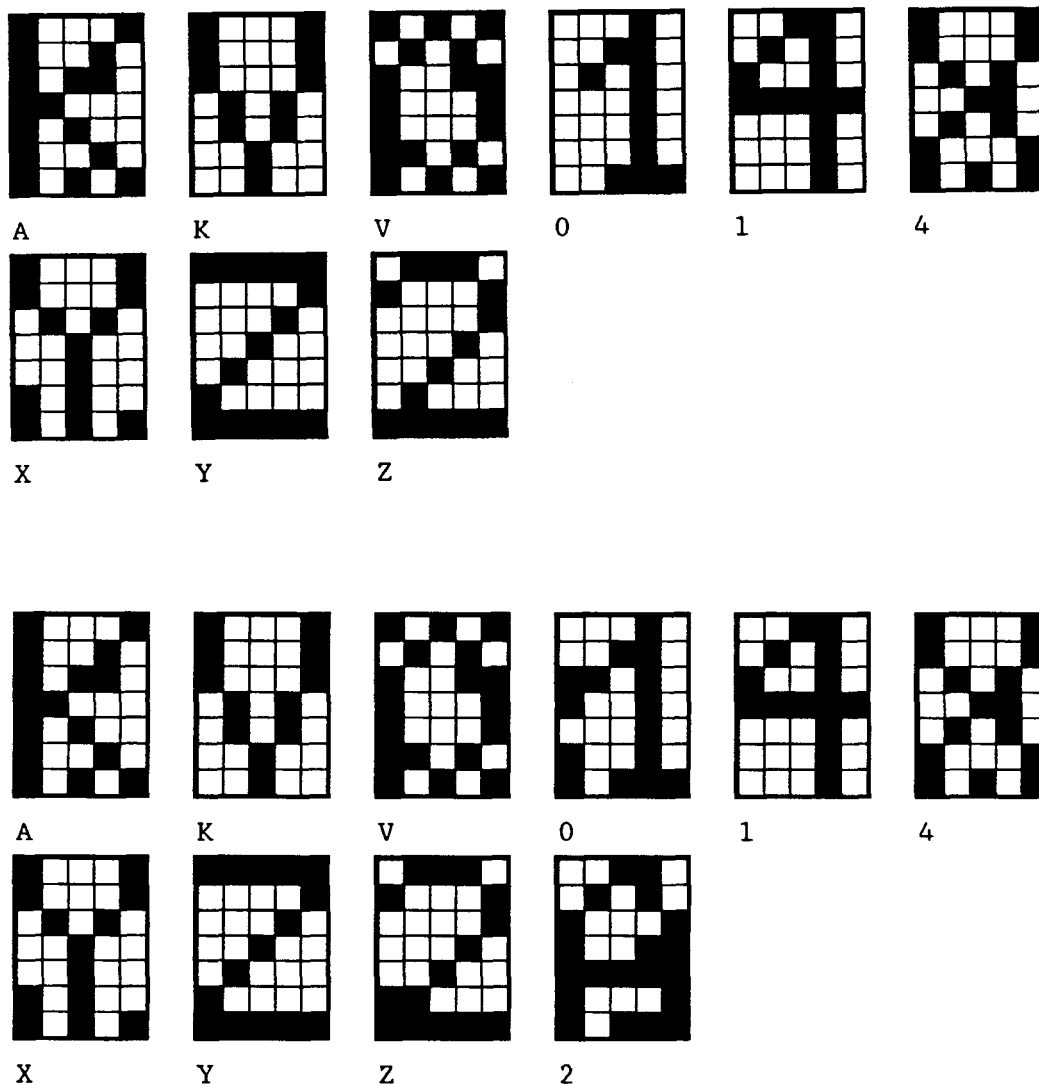


Abbildung IV.25: Ausgelesene Muster a) $z = 9$, b) $z = 10$;

Einsen pro Abschnitt. Palm hat noch eine Reihe von weiteren Kodierungen vorgeschlagen und auf ihre **Tauglichkeit** geprüft. Für praktische Anwendungen ist wichtig, daß solche Kodierungen von der Bedeutung her ähnliche Muster aus der Außenwelt in ähnliche Eingabemuster für den Assoziativspeicher kodieren.

Untersucht werden 4 Kodierungen a) $x = 2$, $l = 18$, $n = 72$; b) $x = 3$, $l = 12$, $n = 96$; c) $x = 4$, $l = 9$, $n = 144$ und d) $x = 5$, $l = 7$, $n = 224$. Tabelle IV.11 zeigt die Ergebnisse für die 4 verschiedenen Kodierungen, wobei zu den Mustern A..Z und 0..9 noch 20 Sonderzeichen gespeichert wurden, d.h. $z = 56$ (siehe Anhang 1). Das kodierte Muster wurde mit dem Originalmuster (als Ausgabemuster) angelernt.

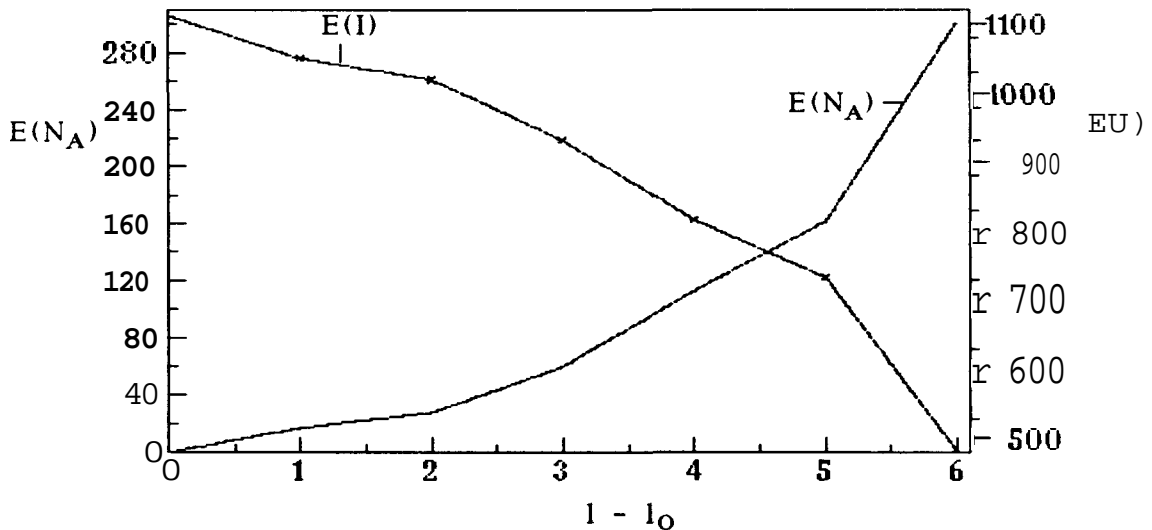


Abbildung IV.27: Gesamte Fehleranzahl $E(N_A)$ und Informationsspeicherkapazität $E(I)$ in Abhängigkeit von der Anzahl $l - l_0$ der fehlenden Einsen in der Eingabe für $n = 224$, $m = 35$, $l = 7$, $k \approx 14.8$. $z = 36$ ASCII-Muster.

Um die 36 Muster wirklich zu speichern, werden die Eingabemuster kodiert und danach auf das Vorgängermuster abgebildet, d.h. $A \rightarrow 9$, $B \rightarrow A$, $C \rightarrow B$, ..., $9 \rightarrow 8$ (Hetero-Assoziation). Als Ausgabemuster wird wieder das Originalmuster ($m = 35$, $k \approx 15$) verwendet. Untersucht wurden die Kodierungen a) $x = 4$, $l = 9$, $n = 144$; b) $x = 5$, $l = 7$, $n = 224$ und c) $x = 6$, $l = 6$, $n = 384$. Für zufällige Muster gelten die Ergebnisse aus Tabelle IV.12.

	a)	b)	c)
Pon EU)	0.624	0.385	0.215
$E(N_t)$	1.14	0.19	0.033
$E(N_A)$	40.99	6.82	1.1725

Tabelle IV.12: Werte des Assoziativspeichers bei verschiedenen Kodierungen für zufällige Muster.

Tabelle IV.13 zeigt die Ergebnisse der Simulation, nachdem alle Muster gespeichert wurden und Tabelle IV.14 die Werte für die Anzahl der nacheinander fehlerfrei speicherbaren Muster. Es zeigt sich, daß, obwohl im Fall c) weniger Fehler für alle 36 Muster gemacht werden, der erste Fehler schon bei 5 Mustern auftritt.

	a)	b)	c)
Pon	0.384	0.223	0.09
E(I)	720.27	807.63	808.42
E(A)	184	139	135
»Einsen	4680	3640	2035
*Kollisionen	2745	1895	836

Tabelle IV.13: Werte des Assoziativspeichers bei verschiedenen Kodierungen für $z = 36$ Muster

	a)	b)	c)
z	11	11	5
Pon	0.198	0.111	0.020
E(I)	327.8	337.8	155
«Einsen	1458	1134	308
»Kollisionen	458	262	29

Tabelle IV.14: Werte des Assoziativspeichers bei verschiedenen Kodierungen für die Anzahl der fehlerfrei speicherbaren Muster.

Als letztes wird der Assoziativspeicher betrachtet, für den auch die Ausgabemuster kodiert sind. Die Eingabemuster werden wieder auf das jeweilige Nachfolgemuster abgebildet. Tabelle IV.15 zeigt die berechneten Werte für zufällige Muster a) $x = 3, l = k = 12, n = m = 96$ und b) $x = 4, l = k = 9, n = m = 144$. In Tabelle IV.16 sind die bei der Simulation erhaltenen Werte für alle $z = 36$ Muster dargestellt mit c) $x = 5, l = k = 7, n = m = 224$; d) $x = 6, l = k = 6, n = m = 384$ und e) $x = 7, l = k = 5, n = m = 640$. In keinem Fall ließen sich alle Muster fehlerfrei speichern. Erhöht man die Kodierung noch weiter, so kommt man irgendwann zu dem Fall, daß die Muster durch die Kodierung orthogonal zueinander stehen, da der Hammingabstand unter den Mustern > 2 ist. Dann lassen sich die Muster fehlerfrei speichern. Der Assoziativspeicher ist allerdings schon für den Fall d) zu groß (siehe Kapitel V).

	a)	b)
Pon	0.433	0.13
E(I)	1757.8	1644.9
$E(N_t)$	0.074	0.0002
$E(N_A)$	2.65	0.0063

Tabelle IV.15: Werte des Assoziativspeichers bei verschiedenen Kodierungen für zufällige Muster

	a)	b)	c)	d)	e)
Pon	0.285	0.096	0.027	0.0076	0.00194
E(I)	890.37	1078	1181.62	1235.6	1243.46
E(N _A)	188	86	33	10	7
*Einsen	5184	2916	1764	1296	900
»Kollisionen	2555	924	395	175	104

Tabelle IV.16: Werte des Assoziativspeichers bei verschiedenen Kodierungen für z = 36 Muster

Die Informationsspeicherkapazität in Tabelle IV.16 ist nicht die mit Gleichung IV.18 berechnet, denn durch die Kodierung der Ausgabemuster trägt ein solches nicht mehr die Information $\text{ld}\binom{m}{k}$, sondern $\text{ld}\binom{2^x}{j} = j * x = m$, d.h.

$$I = \sum_{t=1}^z I_t = \sum_{t=1}^z m - \text{ld}\binom{N_{t+k}}{k}. \quad (\text{IV.51})$$

Zusammenfassung:

In diesem Kapitel wurde die Fehlertoleranz des einfachen Assoziativspeichers dargestellt, und zwar einmal für Störungen bei den Verbindungselementen und zum zweiten für verrauschte Eingabemuster. Dabei zeigt sich, daß der Assoziativspeicher nicht effektiv eingesetzt werden kann, wenn es darauf ankommt, alle Muster fehlerfrei auszulesen. Sind ein oder zwei Bitfehler pro Muster tolerierbar, so erhöht sich die Anzahl der speicherbaren Muster, und der Assoziativspeicher kann geeignet eingesetzt werden. Dies zeigen die zu Anfang des Kapitels dargestellte Theorie und die darauffolgenden Simulationen.

Neu war die für die Technologie interessante Einführung von Verbindungselementen, die nicht nur **ausfallen**, sondern auch durchschalten können, sowie die Ableitungen für verrauschte Muster mit zusätzlichen Einsen in der Eingabe. Beide Ansätze konnten mit Hilfe der Binomial-Verteilung approximativ abgeschätzt werden, wodurch Vorhersagen für große Speicher ermöglicht werden. Für gestörte Verbindungselemente wurde eine **Zuverlässigkeits-Kenngröße** eingeführt, die es erlaubt, die Fehlertoleranz von einfachen Assoziativspeichern mathematisch auszudrücken. Neben der Fehlertoleranz liegt ein weiterer Vorteil des einfachen Assoziativspeichers in seiner regulären Struktur, wodurch sich eine Hardware-Realisierung als neuronaler **ASI-Chip** (Anwendungsorientierte integrierte Schaltung) anbietet. (RÜC89).

V. Adaptive Verfahren zur Bestimmung der Verbindungsgewichte

In Kapitel IV wurde die Verbindungsgewichtsmatrix W iterativ berechnet, indem jede Matrix $\mathbf{W}^{(t)} := \mathbf{I}^{(t)} * \mathbf{O}^{(t)T}$, $\mathbf{I}^{(t)} \in \{0,1\}^n$, $\mathbf{O}^{(t)} \in \{0,1\}^m$ aufaddiert wurde, und zwar durch:

$$\mathbf{W}_0 := \bar{\mathbf{0}} \text{ und}$$

$$\mathbf{W}_i := \mathbf{W}_{i-1} \mathbf{A} \mathbf{W}^{(i)}, \text{ wobei } 0 \text{ die Nullmatrix, } \mathbf{O}^{(t)T} \text{ der transponierte Ausgabevektor und } \mathbf{A} \text{ die konjunktive Verknüpfung von Matrixen ist.}$$

In diesem Kapitel wird die Abbildung $n: F \rightarrow A$ aus Kapitel IV als lineare Abbildung von $\mu: \mathbb{R}^n \rightarrow \mathbb{R}^m$ aufgefaßt. Mit anderen Worten ausgedrückt sucht man eine $m \times n$ Matrix W , so daß:

$$\mathbf{O}^t = W * \mathbf{I}^t \text{ für } t = 1..z \text{ oder } \mathbf{O} = W * \mathbf{I}, \quad (\text{V.1})$$

wobei die Matrizen $/$ und O definiert sind durch:

$$O_{jj} := O_i^j \text{ und } I_{ij} := I_i^j, \text{ d.h. sie bestehen aus den Spaltenvektoren der jeweiligen Mustersätze.}$$

Dieser Ansatz ist ausführlich von T. Kohonen (KOH72, KOH74, KOH84) untersucht worden mit der Intention, eine mathematische Grundlage für den verteilten Korrelationsspeicher zu schaffen. Die in vielen Systemen in der Impulsfrequenz kodierte Information wird in einen analogen Eingabewert transformiert und gespeichert. Der gespeicherte, analoge Ausgabewert kann entsprechend in eine Impulsfolge wieder zurücktransformiert werden. Da jeder analoge Eingabewert durch eine binäre Zahl approximiert werden kann, werden im folgenden nur binäre Eingabevektoren betrachtet. Allgemein kann die Gleichung V.1 exakt gelöst werden, falls die Matrix $/$ invertierbar ist, d.h. wenn die Vektoren \mathbf{I}^t linear unabhängig sind. Es gilt dann:

$$\mathbf{W} = \mathbf{O} * \mathbf{I}^{-1}. \quad (\text{V.2})$$

Falls die Vektoren \mathbf{I}^t nicht linear unabhängig sind, kann W approximativ mit Hilfe von $/ *$, der Pseudoinversen von $/$ berechnet werden, d.h.

$$\mathbf{W} = \mathbf{O} * \mathbf{I}^+. \quad (\text{V.3})$$

Die Pseudoinverse $/ *$ ist gleich der inversen $/^{-1}$, falls $/$ invertierbar ist. Sie existiert für jede Matrix. Die durch $W * \mathbf{I}^t$ berechnete Ausgabevektoren sind bezüglich des euklidischen Abstand zu den Originalmustern minimal.

Zur Berechnung von $/ *$ gibt es viele Verfahren, z. B. das von Albert (ALB72) oder Ben-Israel-Greville (GRE80). Der rekursive Algorithmus von Greville (Greville's Theorem) stellt dabei einen guten Kompromiß zwischen Berechnungseffizienz und einfacher Programmierung dar. Die Idee besteht in der Zerlegung der Originalmatrix in Spalten und der Berechnung der neuen Pseudoinversen unter Hinzunahme eines neuen Spaltenvektors und der bis dahin berechneten Pseudoinversen.

Es bezeichne E die Einheitsmatrix, \mathbf{A}_k eine Matrix A mit k Spalten, die unterteilt wird in $\mathbf{A}_k = [\mathbf{A}_{k-1} | \mathbf{a}_k]$, mit einer Matrix \mathbf{A}_{k-1} die $k-1$ Spalten besitzt.

Das Theorem von Greville besagt:

Satz: V.1:

$$\mathbf{A}_k^+ = \left[\frac{\mathbf{A}_{k-1}(\mathbf{E} - \mathbf{a}_k \mathbf{p}_k^T)}{\|(\mathbf{E} - \mathbf{A}_{k-1} \mathbf{A}_{k-1}^+) \mathbf{a}_k\|^2} \right]$$

$$\mathbf{p}_k = \begin{cases} \frac{(\mathbf{E} - \mathbf{A}_{k-1} \mathbf{A}_{k-1}^+) \mathbf{a}_k}{\|(\mathbf{E} - \mathbf{A}_{k-1} \mathbf{A}_{k-1}^+) \mathbf{a}_k\|^2}, & \text{falls der Nenner } \neq 0, \\ \frac{(\mathbf{A}_{k-1}^+)^T \mathbf{A}_{k-1}^+ \mathbf{a}_k}{1 + \|\mathbf{A}_{k-1}^+ \mathbf{a}_k\|^2} & \text{sonst.} \end{cases}$$

Die Anfangsmatrix \mathbf{A}_1 besteht aus der ersten Spalte von \mathbf{A} . Die Matrix $\mathbf{A}_1^+ = \mathbf{a}_1^T (\mathbf{a}_1^T \mathbf{a}_1)^{-1}$, falls \mathbf{a}_1 nicht der Nullvektor ist, ansonsten ist $\mathbf{A}_1^+ = \mathbf{0}$.

Beispiel V.1:

Als Beispiel soll die die Pseudoinverse \mathbf{A}^+ zur 4 x 3 Matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 7 \\ -5 & 4 & 1 \\ 9 & 3 & -2 \\ 8 & 6 & 10 \end{bmatrix}$$

bestimmt werden. Es ist zunächst

$$\mathbf{A}_1^+ = \begin{bmatrix} 3 \\ -5 \\ 9 \\ 8 \end{bmatrix}^T * \frac{1}{3^2 + (-5)^2 + 9^2 + 8^2} = \frac{1}{179} (3, -5, 9, 8),$$

und somit ergibt sich für $k = 2$

$$\mathbf{a}_2 = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 6 \end{bmatrix}; \quad (\mathbf{E} - \mathbf{A}_1 \mathbf{A}_1^+) \mathbf{a}_2 = \frac{1}{179} \begin{bmatrix} 175 \\ 1021 \\ -12 \\ 586 \end{bmatrix}; \quad \|(\mathbf{E} - \mathbf{A}_1 \mathbf{A}_1^+) \mathbf{a}_2\|^2 = 44.2123$$

$$\mathbf{p}_2 = \frac{1}{7914} \begin{bmatrix} 175 \\ 1021 \\ -12 \\ 586 \end{bmatrix}; \text{ d.h. es ist } \mathbf{A}_2^+ = \frac{\mathbf{A}_1 (\mathbf{E} - \mathbf{A}_1 \mathbf{A}_1^+) \mathbf{a}_2 \mathbf{p}_2^T}{\|(\mathbf{E} - \mathbf{A}_1 \mathbf{A}_1^+) \mathbf{a}_2\|^2}$$

$$= \frac{1}{7914} \begin{bmatrix} 73 & -569 & 402 & 154 \\ 175 & 1021 & -12 & 586 \end{bmatrix};$$

für $k = 3$ erhalten wir nun:

$$\mathbf{a}_3 = \begin{bmatrix} 7 \\ 1 \\ -2 \\ 10 \end{bmatrix}; \quad (\mathbf{E} - \mathbf{A}_2 \mathbf{A}_2^+) \mathbf{a}_3 = \frac{1}{7914} \begin{bmatrix} 37104 \\ 21216 \\ 46320 \\ 24936 \end{bmatrix}; \quad \|(\mathbf{E} - \mathbf{A}_2 \mathbf{A}_2^+) \mathbf{a}_3\|^2 = 73.35254$$

$$p_3 = \frac{1}{580512} \begin{bmatrix} 37104 \\ -21216 \\ -46320 \\ 24936 \end{bmatrix}; \quad \text{d.h. es ist } \mathbf{A}_3^* = \begin{bmatrix} \mathbf{A}_2 (E - \mathbf{a}_3 \mathbf{p}_3^T) \\ \mathbf{p}_3^T \end{bmatrix}^n$$

$$= \frac{1}{580512} \begin{bmatrix} 2176 & -39920 & 33456 & 9160 \\ -25280 & 96688 & 46704 & 17368 \\ 37104 & -21216 & -46320 & 24936 \end{bmatrix};$$

Für die Berechnung der Verbindungsgewichte in diesem Kapitel werden wieder die ASCII-Kodierungen als Eingabevektoren \mathbf{I} verwendet (siehe Abschnitt IV.12 und Anhang 1) und zu einer Matrix $/$ (s.o.) zusammengefaßt. Zu $/$ wird die Pseudoinverse $/^*$ berechnet und mit der Matrix O multipliziert. Für die Anzahlen $z = 5, 10, 15, 20, 25, 30, 35, 40, 45$ (Auto-Assoziation) sind in Abbildung V.1 die durch die Multiplikation entstandenen Matrizen W als 3D-Plot dargestellt. Man erkennt deutlich, daß sich die Matrix W zur Einheitsmatrix entwickelt ($z = 35$). Die Einheitsmatrix wird wieder zerstört, für $z > m = n$. Der Wertebereich der Verbindungsgewichte wird immer größer, je mehr Muster gespeichert werden. Damit empfiehlt sich das Verfahren nicht, wenn die Gewichte analog realisiert werden.

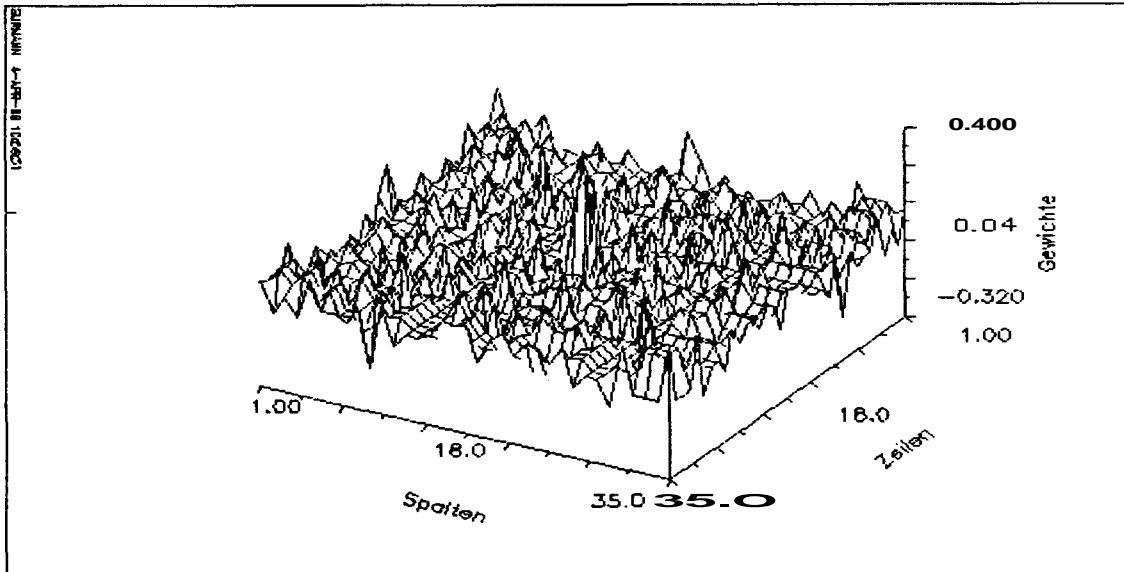
Abbildung V.2 zeigt die Matrix W der Verbindungsgewichte für den Hetero-Assoziationsfall. Die Hetero-Assoziation wurde erreicht, indem jedes Muster auf sein Nachfolgemuster abgebildet wurde ($A \rightarrow B, B \rightarrow C, \dots$). Der Wertebereich der Verbindungsgewichtsmatrix wird noch größer als im Auto-Assoziationsfall. Er steigt explosionsartig, wenn die Anzahl der Muster die Dimension der Verbindungsmatrix übersteigt (Abbildung V.2: $z = 40, 45$). Für $z \leq 35$ lassen sich alle Muster fehlerfrei auslesen, wobei zwei verschiedene Ausleseverfahren verwendet wurden. Zum einen wurde der Ausgabevektor \mathbf{O}_1^t berechnet, indem der durch $\mathbf{A}^t = W * \mathbf{I}^t$ erhaltene Aktivierungsvektor \mathbf{A}^t mit Hilfe der Ausgabefunktion $\delta_1: \mathbb{R}^m \rightarrow \mathbb{R}^m$,

$$\delta_1(\mathbf{A}) := \begin{cases} 1, & \text{falls } \mathbf{a}_i \in [0.9, 1.1], \\ 0, & \text{falls } \mathbf{a}_i \in [-0.1, 0.1], \\ \mathbf{a}_j & \text{sonst,} \end{cases} \quad \forall i \in \{1..m\}$$

berechnet wurde. Die Ausgabefunktion δ_1 'glättet' den Aktivierungsvektor, bei dem durch numerische Berechnungen Werte nahe an eins bzw. nahe an 0 entstanden sind. Zum anderen wurde der binäre Ausgabevektor \mathbf{O}_2 berechnet durch die Schwellenfunktion $\delta_2: \mathbb{R}^m \rightarrow \{0,1\}^m$ mit

$$\delta_2(\mathbf{A}) := \begin{cases} 1, & \text{falls } \mathbf{a}_j \geq \text{TH} \\ 0, & \text{sonst.} \end{cases} \quad \forall i \in \{1..m\}$$

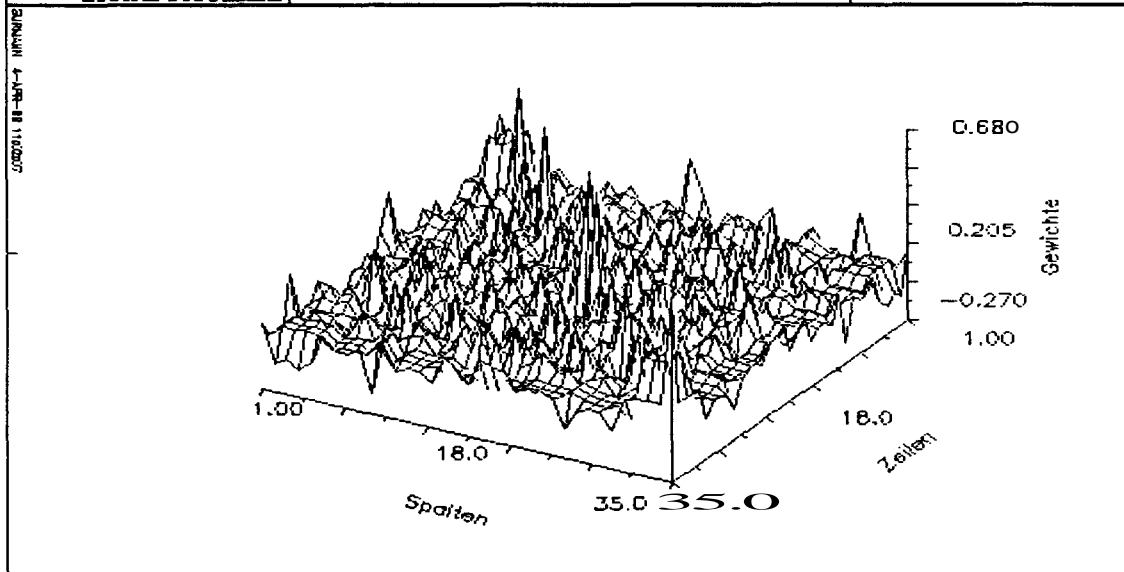
Wird $\text{TH} = \mathbf{a}_j$ für $j = 1..n$ nacheinander gewählt, so erhält man den Ausgabevektor mit minimalem Fehler. Die so erhaltene Fehleranzahl $E(N_A)$ wird im folgenden stets dargestellt. Tabelle V.1 zeigt die Fehleranzahl $E(N_A)$ für $z = 40$ und 45 im Auto- und im Hetero-Assoziationsfall.



BE Bauelemente
der
Elektrotechnik

5 Auto-Assoziation

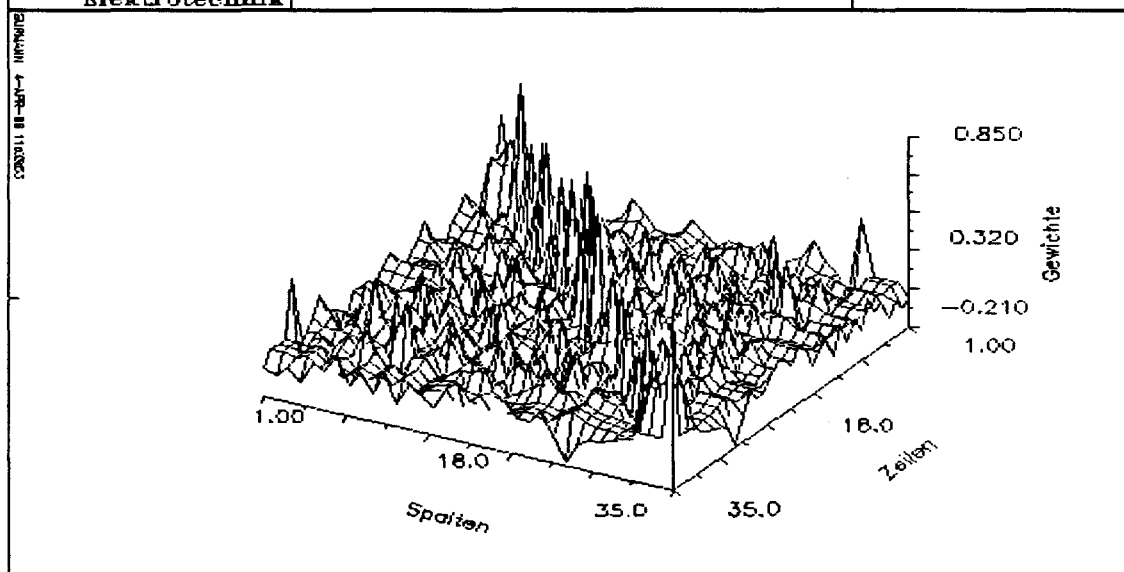
3 D-Plot



BE Bauelemente
der
Elektrotechnik

10 Auto-Assoziation

3 D-Plot

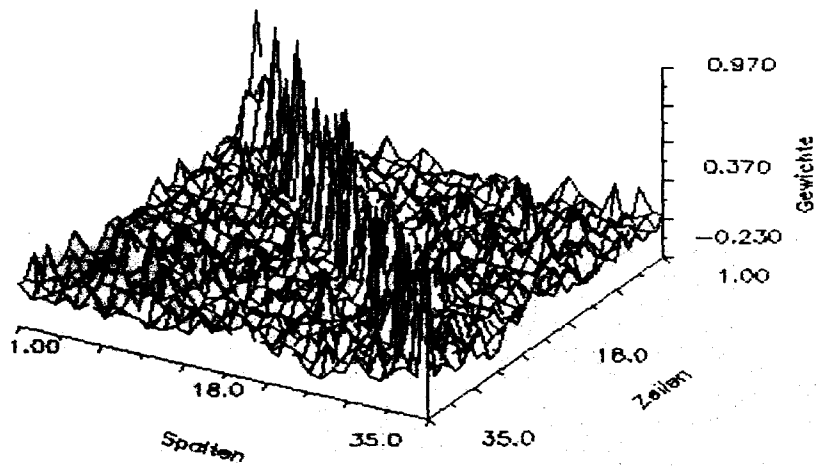


BE Bauelemente
der
Elektrotechnik

15 Auto-Assoziation

3 D-Plot

320011 BE-447-7 NINWINE

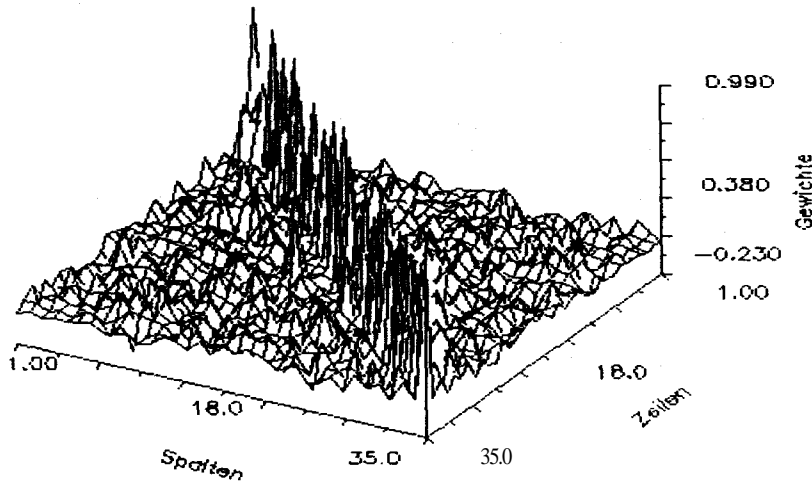


BE Bauelemente
der
Elektrotechnik

20 Auto-Assoziation

3 D-Plot

320011 BE-447-7 NINWINE

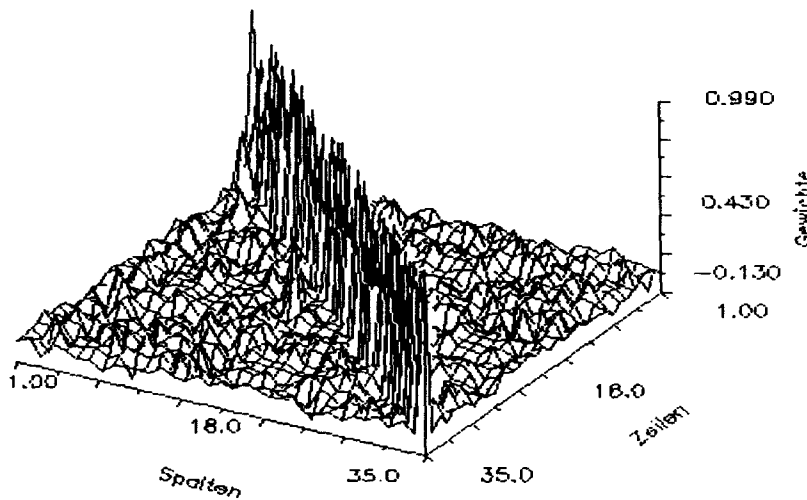


BE Bauelemente
der
Elektrotechnik

25 Auto-Assoziation

3 D-Plot

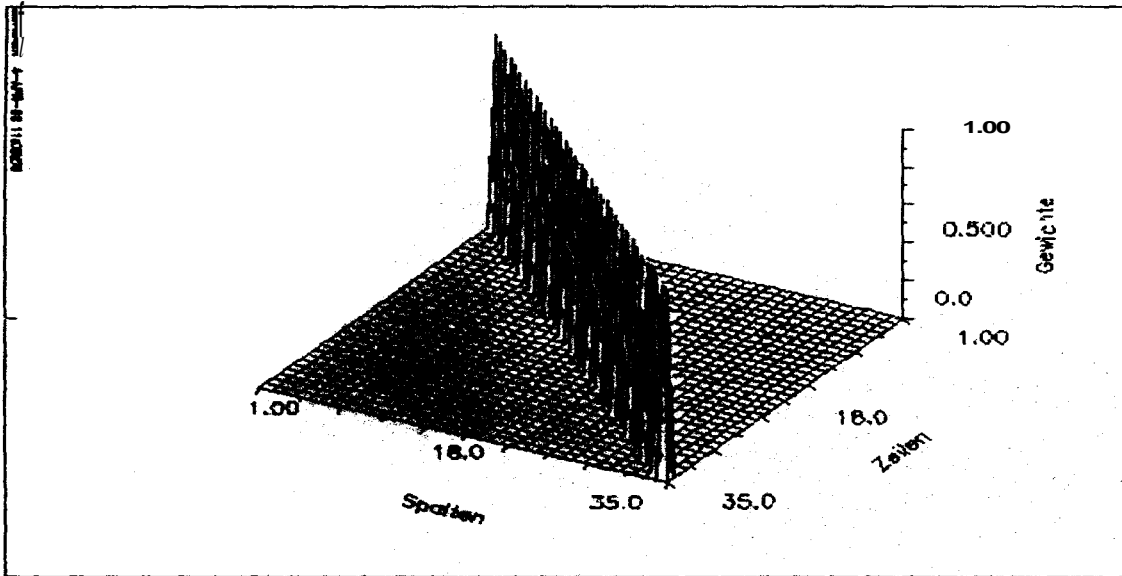
320011 BE-447-7 NINWINE



BE Bauelemente
der
Elektrotechnik

30 Auto-Assoziation

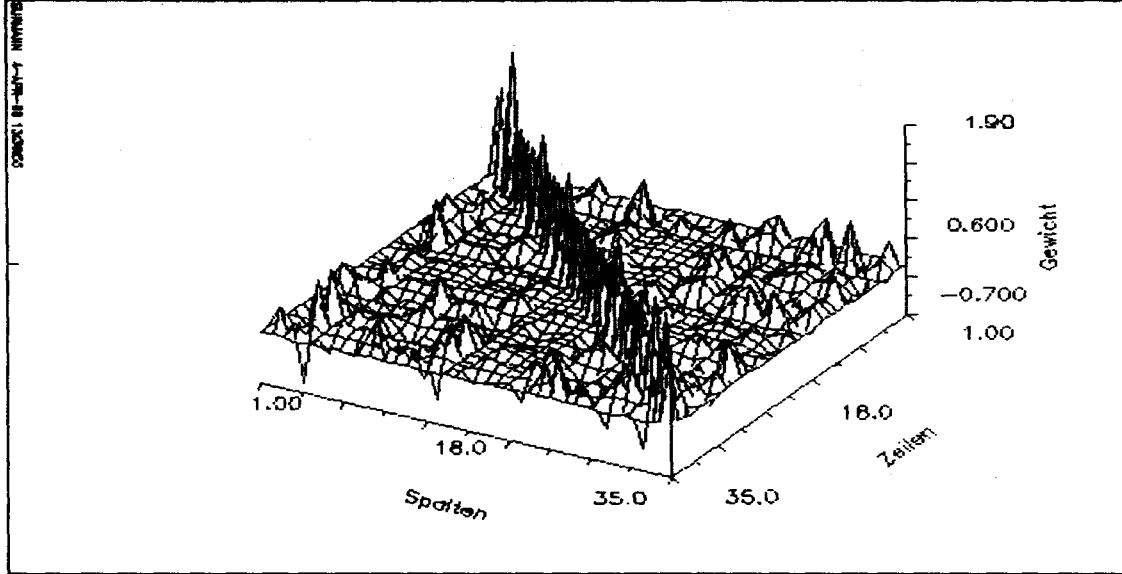
3 D-Plot



BE Bauelemente
der
Elektrotechnik

35 Auto-Assoziation

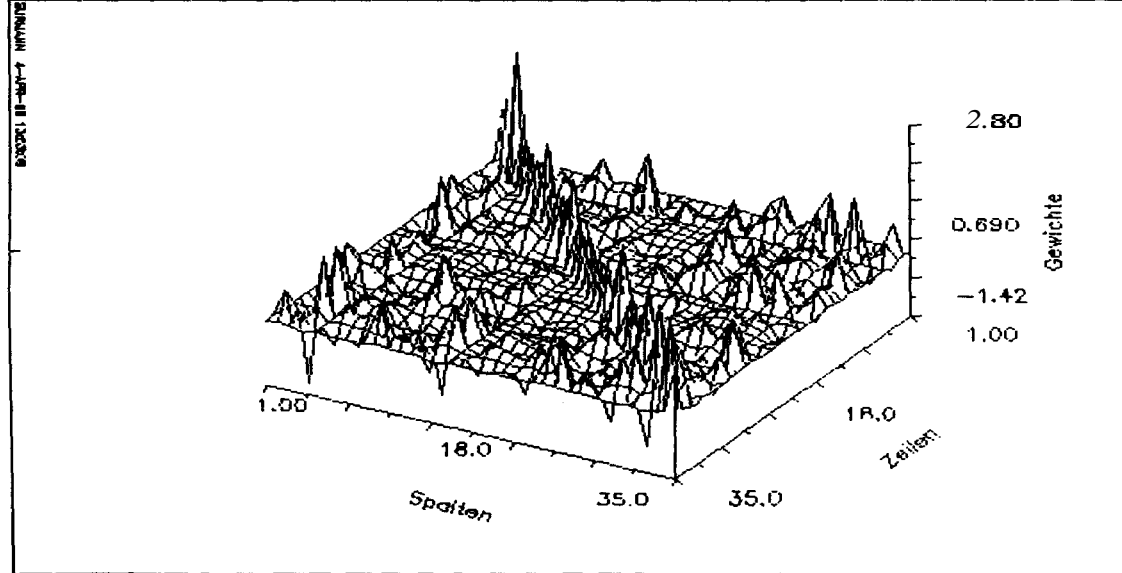
3 D-Plot



BE Bauelemente
der
Elektrotechnik

40 Auto-Assoziation

3 D-Plot



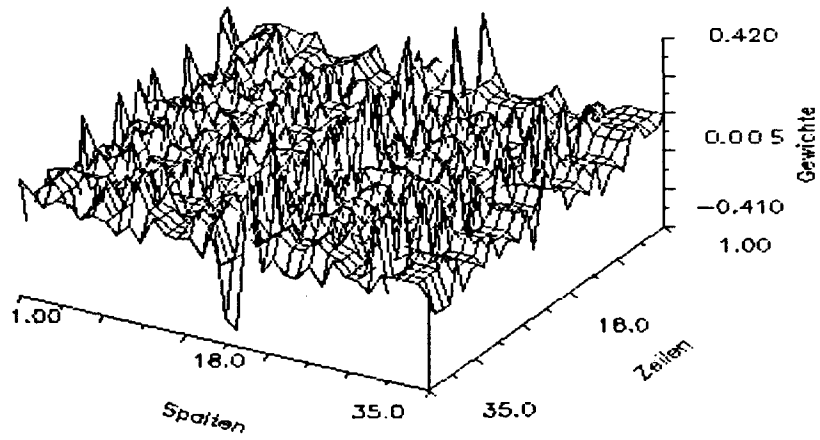
BE Bauelemente
der
Elektrotechnik

45 Auto-Assoziation

3 D-Plot

Abbildung V.1: Verbindungsgewichte für verschiedene Musteranzahlen

620111 BE-BA-7 INFORME

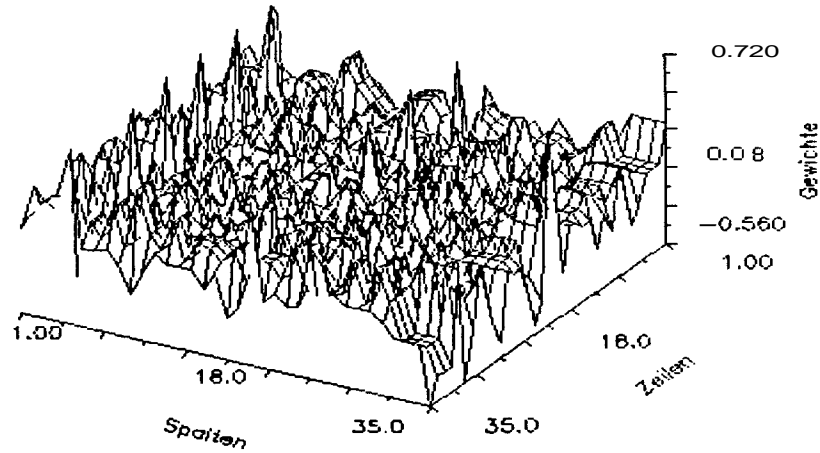


BE Bauelemente
der
Elektrotechnik

5 Hetero-Assoziation

3 D-Plot

620111 BE-BA-7 INFORME

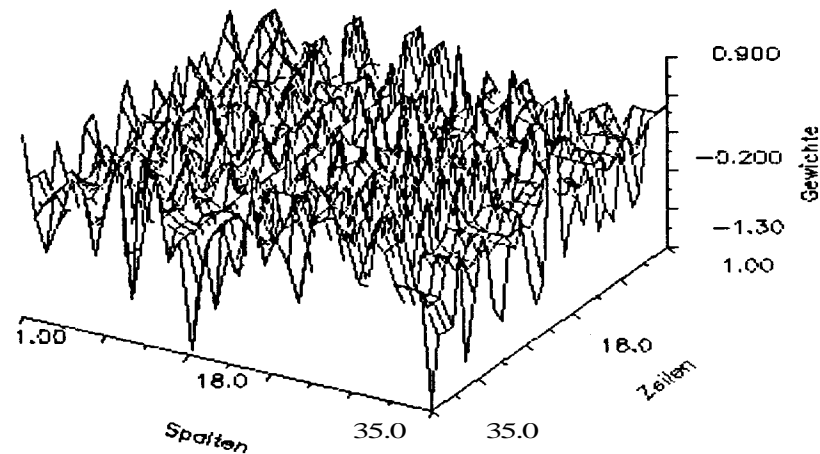


BE Bauelemente
der
Elektrotechnik

10 Hetero-Assoziation

3 D-Plot

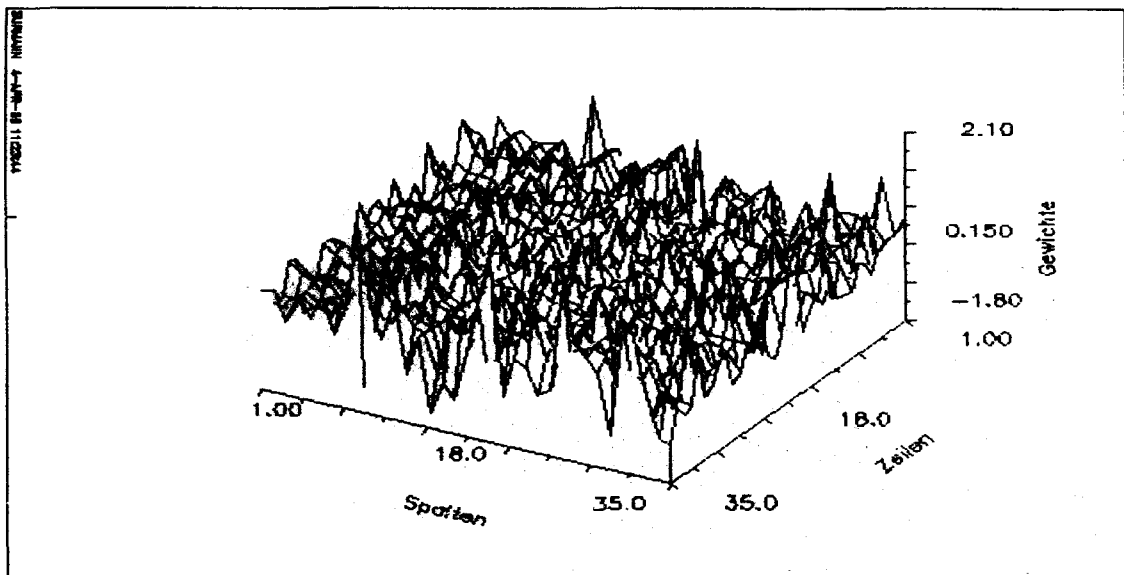
620111 BE-BA-7 INFORME



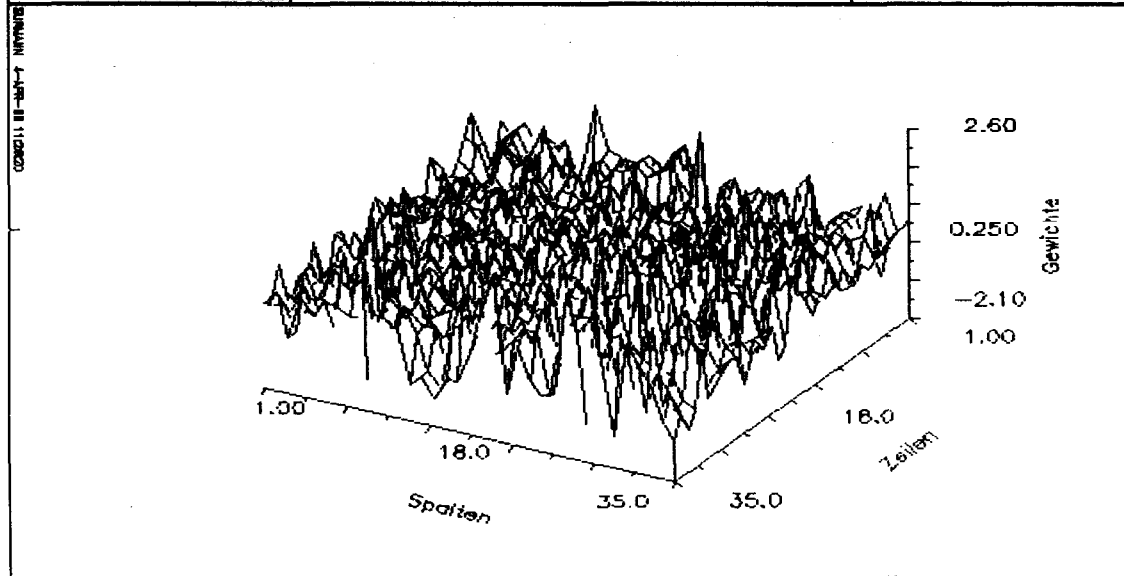
BE Bauelemente
der
Elektrotechnik

15 Hetero-Assoziation

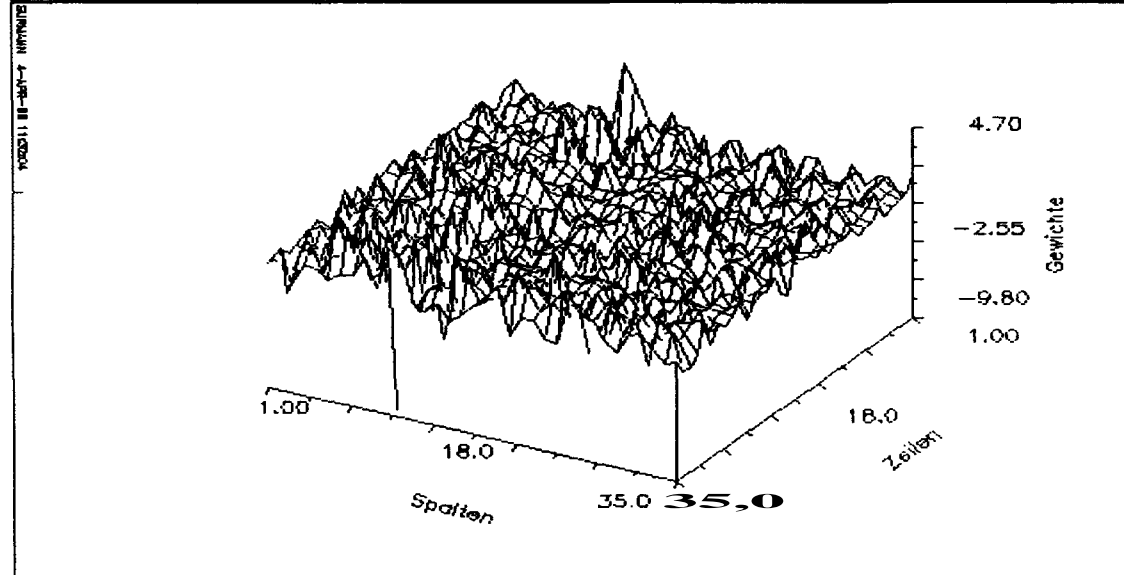
3 D-Plot



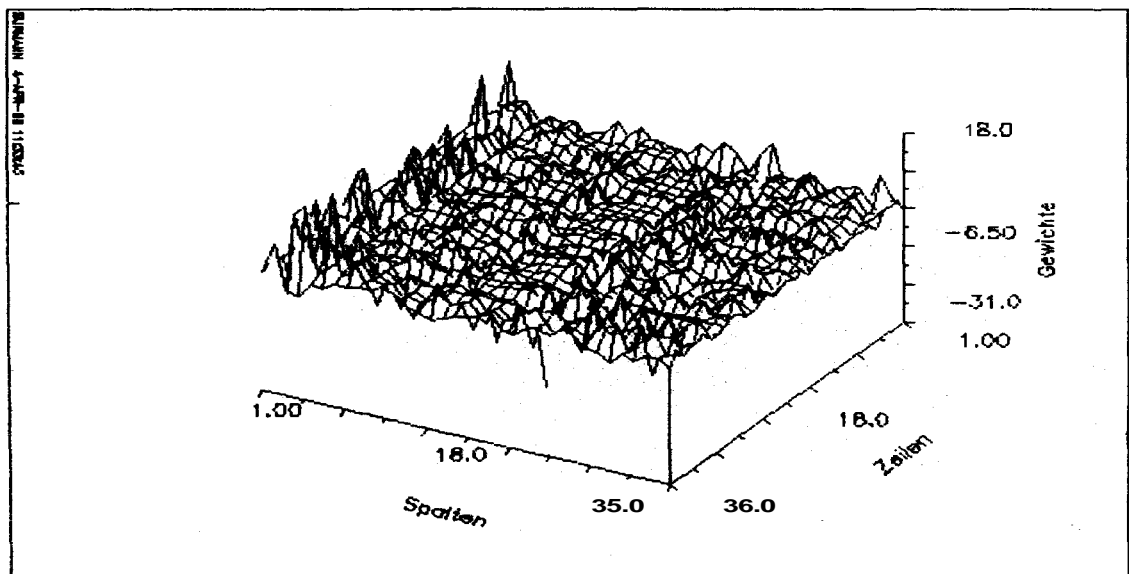
BE Bauelemente der Elektrotechnik | 20 Hetero-Assoziation | 3 D-Plot



BE Bauelemente der Elektrotechnik | 25 Hetero-Assoziation | 3 D-Plot



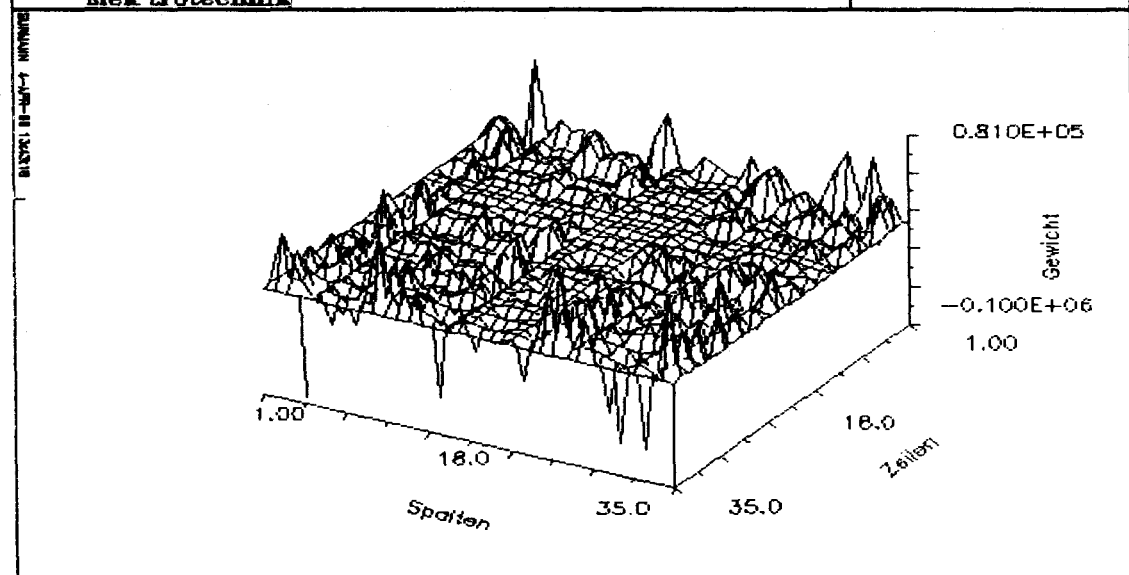
BE Bauelemente der Elektrotechnik | 30 Hetero-Assoziation | 3 D-Plot



BE Bauelemente
der
Elektrotechnik

3S Hetero-Assoziation

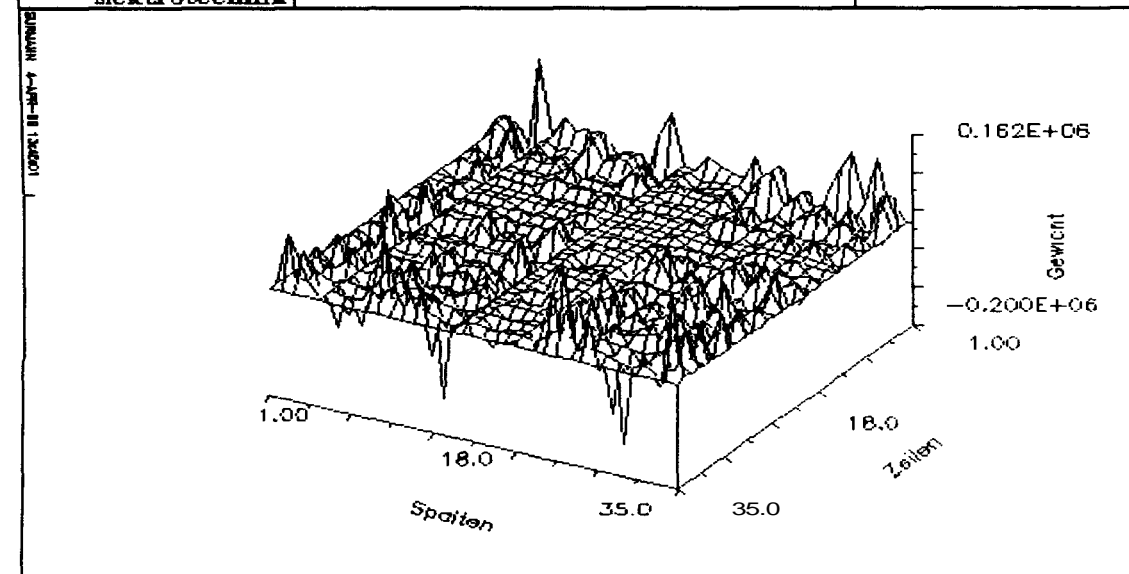
3 D-Plot



BE Bauelemente
der
Elektrotechnik

40 Hetero-Assoziation

3 D-Plot



BE Bauelemente
der
Elektrotechnik

45 Hetero-Assoziation

3 D-Plot

Abbildung V.2: Verbindungsgewichte bei verschiedenen Musteranzahlen

Das Fehlermaß $E(N_A)$ entspricht der Summe der euklidischen Abstände jedes berechneten Ausgabemusters zum Originalmuster.

z	$E(N_A)_{\text{Auto}}$	$E(N_A)_{\text{Hetero}}$
40_{δ_1}	34.95	3265857
40_{δ_2}	5	465
45_{δ_1}	87.46	7885978
45_{δ_2}	83	512

Tabelle V.1: Gesamte Fehleranzahl bei $z = 40$ bzw. $z = 45$ Mustern unter Verwendung der Ausgabefunktionen δ_1 und δ_2 .

V.1 Simulation mit **gestörten** Eingabevektoren

Für eine Simulation mit gestörten Eingabevektoren werden für die ersten Simulationsexperimente die erste und die letzte Zeile der 5×7 - Punktmatrizen auf Null gesetzt, d.h. die Eingabevektoren \mathbf{I}^t enthalten an den Positionen $i = 5$ bis $i = 30$ korrekte Werte. In Abbildung V.3 ist für die ersten 17 Muster als Fehlermaß der euklidische Abstand EA der vom Netz berechneten Muster zum Originalmuster dargestellt. Die grauen Balken entsprechen darin der Bitfehleranzahl $E(N_t)$, denn die Ausgabevektoren wurden mit Hilfe der Ausgabefunktion δ_2 berechnet. Die Zahlen entsprechen der Numerierung der Buchstaben ($A=1, B=2, \dots$).

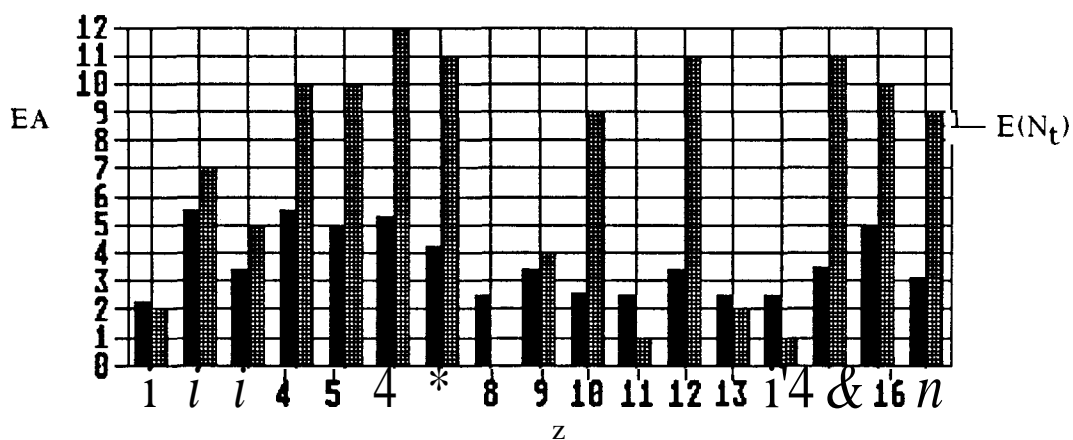


Abbildung V.3: Der euklidische Abstand EA des berechneten Musters zum Originalmuster ($z = 17$). Hetero-Assoziation ($A \hat{=} 1, B \hat{=} 2 \dots$). Die grauen Balken sind mit der Ausgabefunktion δ_2 und die schwarzen Balken mit der Ausgabefunktion δ_1 berechnet.

Abbildung V.4 zeigt den euklidischen Abstand EA für alle 35 Muster. Es zeigt sich, daß die Fehlertoleranz geringer ist als für $z = 17$ Muster, da die Muster weiter vom Originalmuster entfernt liegen.

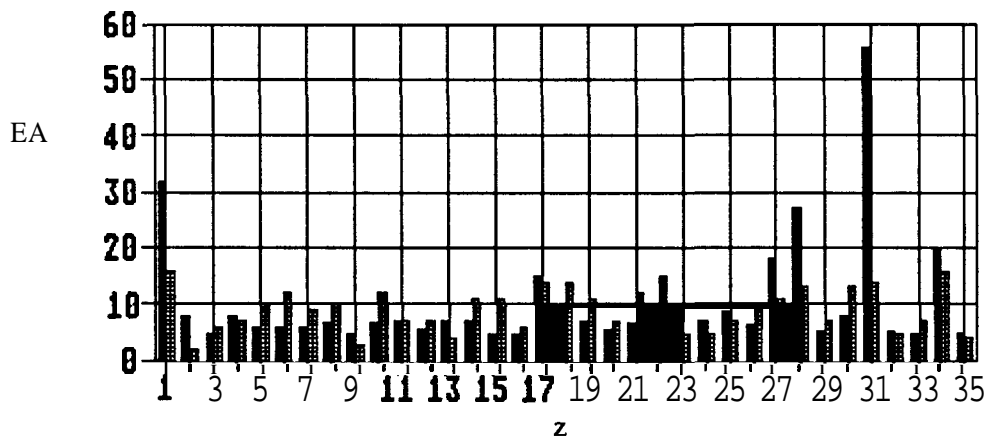


Abbildung V.4: Der euklidische Abstand EA des berechneten Musters zum Originalmuster ($z = 35$). Hetero-Assoziation ($A \hat{=} 1, B \hat{=} 2\dots$). Die grauen Balken sind mit der Ausgabefunktion δ_2 und die schwarzen Balken mit der Ausgabefunktion δ_1 berechnet.

Abbildung V.5 verallgemeinert die beiden vorherigen Abbildungen und zeigt die Summe der euklidischen Abstände für alle gespeicherten Muster dividiert durch die Anzahl der Muster z für die Ausgabefunktionen δ_1 und δ_2 in Abhängigkeit von der Anzahl der gespeicherten Muster. Dabei zeigt sich, daß bei Fortlassen der ersten und der letzten Zeile der (5×7) ASCII-Punktmatrizen, nur noch die ersten fünf Muster fehlerfrei gespeichert werden können und die ersten sieben Muster zu weniger als zwei Bitfehler führen. An dieser Abbildung zeigt sich deutlich die sinkende Fehlertoleranz bei gestörten Eingabemustern, je mehr Muster gespeichert werden.

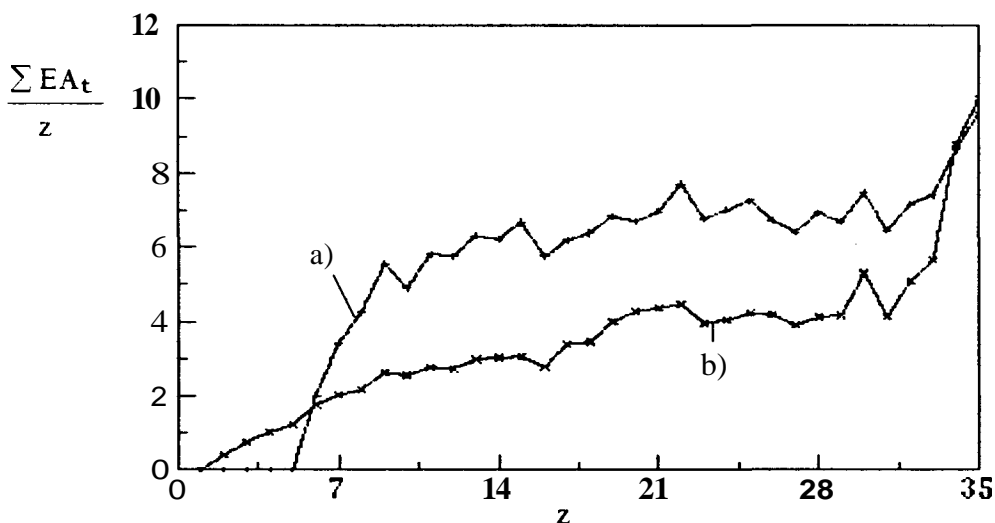


Abbildung V.5: Durchschnittlicher euklidischer Abstand EA aufgetragen für alle Muster a) unter Verwendung der Ausgabefunktion δ_1 und b) der Ausgabefunktion δ_2 .