Topologie- und Dimensionserhaltung bei selbstorganisierenden künstlichen Neuronalen Netzen

Diplomarbeit

vorgelegt von

Erik Vellmete

aus Kassel

angefertigt im
Institut für Numerische und Angewandte Mathematik
der Georg-August-Universität zu Göttingen
1995

Inhaltsverzeichnis

1. Einleitung **2**

2. Selbstorganisierende künstliche Neuronale Netze - Kohonens Modell **6**
- 2.1 Abriß eines biologischen Modells eines Neurons . . . 6
- 2.2 Definition eines selbstorganisierenden künstlichen Neuronalen Netzes . . . 6
- 2.3 Der Algorithmus für ein selbstorganisierendes Netz . . . 8
- 2.4 Unitabstand und Nachbarschaftsfunktion . . . 9
- 2.5 Beispiele . . . 10
- 2.6 Untersuchung der Fehlerfunktion . . . 11
- 2.7 Steuerung der Parameter ϵ und σ . . . 15

3. Nachbarschaftserhaltung **17**
- 3.1 Quantifizierung der Nachbarschaftserhaltung . . . 17
  - 3.1.1 Definition der nächsten Nachbarn . . . 18
  - 3.1.2 Ein Maß für die Nachbarschaftserhaltung . . . 18
- 3.2 Dimensionserkennung . . . 21
- 3.3 Beispiele zur Dimensionserkennung . . . 23
- 3.4 Untersuchung der Kurven . . . 25
- 3.5 Weitere Beispiele zur Dimensionserkennung . . . 26

4. Nicht eindeutige Reihenfolge der Unitnachbarn **30**
- 4.1 Invarianten bei einer Änderung der Reihenfolge . . . 30
- 4.2 Die mögliche Abweichung für eine Sequenz . . . 31
- 4.3 Schranken für die mögliche Abweichung . . . 32
- 4.4 Untersuchung der Beispiele aus 3.3 und 3.5 . . . 35
- 4.5 Vorschlag für eine Lösung des Problems der nicht eindeutigen Reihenfolge 39

5. Zusammenfassung **41**

1. Einleitung

Das menschliche Gehirn ist einem digitalen Rechner in vielerlei Hinsicht überlegen. So erkennt schon ein einjähriges Baby Gesichter und Gegenstände wesentlich schneller und besser, als ein sehr komplexes Programm, das auf einem Supercomputer läuft. Der Computer ist nur bei primitiven Rechenaufgaben wesentlich schneller als das Gehirn. Dieses hat dagegen dem Computer mehrere Eigenschaften voraus:

- Es ist flexibel. Durch „Lernen“ paßt sich das Gehirn einer neuen Aufgabe an, oft ohne daß der Mensch aktiv etwas dafür tun muß.
- Es verarbeitet ungenaue und inkonsistente Daten.
- Es ist sehr fehlertolerant. Täglich absterbende Zellen beeinträchtigen die Gesamtfunktion nur unwesentlich.
- Es arbeitet hochgradig parallel.

Die *Neuroinformatik* beschäftigt sich damit, die eben genannten Eigenschaften für Computer verfügbar zu machen. Dabei werden grundlegende Erkenntnisse aus der Funktionsweise des Gehirns zur Entwicklung neuartiger Hardware (massiv parallele Netzwerke, Neuro-Chip) und Software (künstliche Neuronale Netze) benutzt. Mit Hilfe künstlicher Neuronaler Netze werden Probleme aus sehr unterschiedlichen Bereichen erfolgreich bearbeitet: zum Beispiel Bildverarbeitung (Erkennen von Gegenständen), Steuerungstechniken (Robotersteuerung) und nichtlineare Optimierung (Travelling Salesman Problem).

In dieser Arbeit wird eine Möglichkeit dargestellt und genauer untersucht, mit *selbstorganisierenden künstlichen Neuronalen Netzen* die „Dimension“ einer Datenmenge „herauszufinden“. Was ist die „Dimension“ einer Datenmenge und wozu ist es nützlich, diese herauszufinden? Zur Beantwortung der zweiten Frage zunächst ein Beispiel: Ich betrachte 50 Geräte, mit denen Wetterdaten (Temperatur, Luftdruck, Windgeschwindigkeit, …) gemessen werden. Die Daten lassen sich als fünfzigdimensionale Vektoren darstellen. Allerdings sind sie im allgemeinen stark korreliert. Zur weiteren Verarbeitung (Wetterprognose) ist es sinnvoll, die „wirkliche Dimension“ der Daten zu kennen, das heißt Korrelationen zu entdecken und zu eliminieren.

Um die „Dimension“ von Daten herauszufinden, werden diese mit Netzen verschiedener Dimension verbunden. Dabei sind zunächst keine Neuronalen Netze gemeint. Ein eindimensionales Netz ist ein Polygonzug, ein zwei- und ein dreidimensionales Netz sind in Fig. 1.1 und 1.2 dargestellt. Höherdimensionale Netze werden analog konstruiert. Die „Dimension“ einer Datenmenge ist durch die eines Netzes bestimmt, mit dem sich die Daten „gut“ verbinden lassen. Tritt einer der folgenden „Defekte“ auf, so ist das Netz nicht für diese Daten geeignet:

1. Es gibt Verbindungslinien des Netzes, die sich schneiden.
2. Das Netz ist gefaltet. (Eine Peano-Kurve ist ein Beispiel für ein eindimensionales, gefaltetes Netz.)

3. Das Netz ist verzerrt, das heißt die Abstände zwischen verbundenen Knoten sind sehr unterschiedlich.

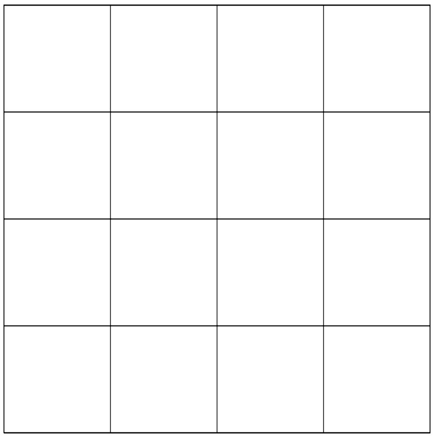


Fig. 1.1: ein zweidimensionales Netz mit 25 Knoten

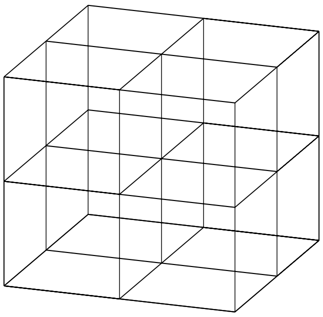


Fig. 1.2: ein dreidimensionales Netz mit 27 Knoten

Um diesen Dimensionsbegriff mit endlich vielen Knoten auch auf unendliche Mengen anwenden zu können, wird statt einer Verbindung der Daten mit einem Netz (diese benötigt ebenso viele Netzknoten wie Datenvektoren) eine Überdeckung mit einem Netz betrachtet, wobei die Anzahl der Knoten frei gewählt werden kann. Eine „gute“ Überdeckung ist dadurch charakterisiert, daß „in der Nähe“ jedes Datenvektors ein Netzknoten liegt und umgekehrt. Damit gibt es neben 1 – 3 einen weiteren möglichen Defekt:

4. Der Datenraum wird nicht vollständig überdeckt oder das Netz liegt nicht „nahe“ an den Daten.

Zusammengefaßt ergibt sich folgendes Dimensionskriterium: Die „Dimension“ einer Datenmenge ist durch die Dimension eines Netzes bestimmt, mit dem sich der Datenraum möglichst ohne Auftreten eines der Defekte 1 – 4 überdecken läßt. Es ist mit den bisher betrachteten Netztypen nicht immer möglich, einen Datenraum zu überdecken, ohne daß einer der Defekte auftritt. In der einschlägigen Literatur wird der problematische Begriff der „Dimension“ einer Datenmenge nicht erwähnt. An zwei einfachen Beispielen möchte ich das Dimensionkriterium verdeutlichen:

1. Sei X die Menge von 10.000 zufällig aus $[0,1]^2$ gewählten Punkten. Welche „Dimension“ hat X? Sicherlich läßt sich eine (eindimensionale) Kurve durch alle Punkte legen, die sich nicht selbst schneidet. Diese Kurve ist aber immer gefaltet. Dagegen lassen sich die Punkte „zweidimensional“ (wie in Fig. 1.1) überdecken, ohne daß es zu einer Faltung kommt. Im Fall einer nicht mehr leicht vorstellbaren „dreidimensionalen“ Überdeckung der Punkte (wie in Fig. 1.2) lassen sich Überschneidungen von Verbindungslinien nicht verhindern. Deshalb möchte ich davon sprechen, daß X die „Dimension“ 2 hat. Statt der 10.000 Punkte kann mit demselben Ergebnis auch $X = [0,1]^2$ gewählt werden.

2. Welche Dimension hat eine Kugeloberfläche? Eine Kurve kann eine Sphäre nur durch Faltung vernünftig überdecken. Ein zweidimensionales Netz läßt sich nur so auf eine Sphäre legen, daß es entweder zu Überschneidungen der Verbindungslinien kommt oder daß ein Teil der Spähre nicht überdeckt wird oder daß das Netz stark verzerrt wird. Bei einem dreidimensionalen Netz treten ebenfalls Probleme auf: Entweder liegen nur die „äußeren“ oder auch die „mittleren“ Knoten nahe der Sphäre. In beiden Fällen kommt es zu starken Verzerrungen des Netzes. Mit den bisher betrachteten Netzarten kommt es also immer zu Defekten bei der Überdeckung einer Sphäre.
   Wird dagegen ein geschlossenes, zweidimensionales Netz gewählt, bei dem die vier Ränder zu einem Punkt identifiziert werden, so läßt sich die Sphäre sehr gut überdecken. Geschlossene Netze werden im folgenden nicht betrachtet.

Nun zu selbstorganisierenden künstlichen Neuronalen Netzen. Die bisher betrachteten Netze haben einiges mit diesen zu tun. Bei der üblichen graphischen Darstellung der Neuronalen Netze entstehen die eben betrachteten „normalen“ Netze. Dabei befinden sich die „Neuronen“ in den Knoten. Der Algorithmus für ein selbstorganisierendes künstliches Neuronales Netz ist vereinfacht gesagt so konzipiert, daß dieses die ihm präsentierten Daten so gut wie möglich überdeckt. Das bedeutet, daß möglichst keiner der Defekte 1 – 4 auftritt.

Durch Überprüfung eines Netzes kann festgestellt werden, ob dieses einen der eben genannten Defekte aufweist. Durch Quantifizierung der auftretenden Defekte ist es möglich, verschiedene Netze zu vergleichen. Die „Dimension“ einer Datenmenge ist dann durch die Dimension des künstlichen Neuronalen Netzes bestimmt, das unter den getesteten den kleinsten Defekt aufweist. Mit einem Netz dieser Dimension läßt sich die Datenmenge am besten repräsentieren.

Als ein Anwendungsbeispiel für selbstorganisierende künstliche Neuronale Netze möchte ich kurz ein Grundproblem der Robotersteuerung vorstellen: die Positionierung eines Endeffektors im Raum (zum Greifen, Schweißpunkte anbringen, ...). Es wird einschränkend angenommen, daß es eine eindeutige Beziehung zwischen der Position des Endeffektors (x, y, z) und der Stellung $\Theta = (\Theta_1, \Theta_2, \Theta_3)$ der drei Robotergelenke gibt. Auf den Raum, in dem der Endeffektor positioniert werden soll, sind zwei Kameras gerichtet. Ein Punkt des Raumes erzeugt in der Bildebene jeder Kamera einen zweidimensionalen Lagevektor. Beide Vektoren ergeben zusammengefaßt einen vierdimensionalen Vektor. Dieser bestimmt den Punkt eindeutig, da das Arbeitsgebiet eine dreidimensionale Untermannigfaltigkeit in $I\!R^4$ ist.

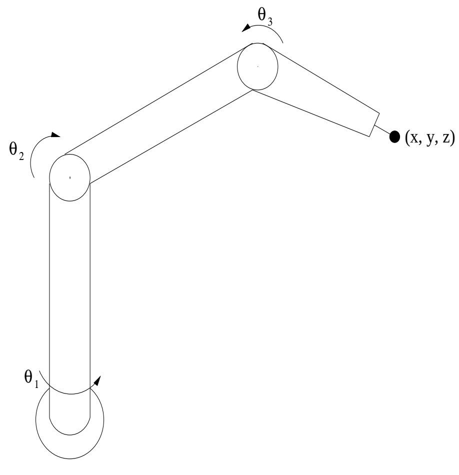


Fig. 1.3: ein Roboterarm mit drei Gelenken

Stellen wir uns nun ein zufällig im Arbeitsgebiet des Endeffektors verteiltes dreidimensionales Netz vor. In jedem Knoten r des Netzes sind neben der Position w_r des Knotens zwei weitere Größen gespeichert: ein dreidimensionaler Vektor Θ_r zur Positionierung des Endeffektors und eine 3×4 Matrix A_r zur Korrektur, falls die Positionierung nicht korrekt erfolgt.

Das Verfahren, mit dem das Netz „trainiert“ wird, ist zu umfangreich, um hier dargestellt zu werden. Hier eine stark vereinfachte Version:

1. Wähle zufällig einen Zielpunkt im Arbeitsgebiet und bestimme mit den Kameras den zugehörigen Vektor u.

2. Bestimme den Knoten r', der am nächsten an u liegt.

3. Verschiebe den Knoten r', so daß er näher an u liegt, und verringere den Defekt des Netzes (dabei werden auch Knoten in der „Nähe“ von r' verschoben).

4. Bringe den Endeffektor in eine Zwischenposition durch Einstellen der Gelenkwinkel auf
$$\Theta_i = \Theta_{r'} + A_{r'}(u - w_{r'})$$
und bestimme mit den Kameras den zugehörigen Vektor v_i.
($A_{r'}$ ist eine Approximation an die Jakobimatrix der Transformation $\Theta_i(u)$ an der Stelle $w_{r'}$ des Knotens r'. Die Einbeziehung der Matrix $A_{r'}$ ermöglicht es, auch Zielpunkte in der „Nähe“ der zu $\Theta_{r'}$ gehörenden Position zu erreichen.)

5. Korrigiere die Position v_i gemäß
$$\Theta_f = \Theta_{r'} + A_{r'}(u - v_i)$$
und bestimme mit den Kameras die Endposition v_f des Endeffektors.

6. Verändere $\Theta_{r'}$ und $A_{r'}$ so, daß bei erneuter Präsentation desselben Zielpunktes v_f näher an u liegt.

Das Verfahren wird ausführlich in [1], S. 171–209 besprochen.

Es kommt zu folgenden Ergebnissen: Das anfangs völlig verknotete Netz wird entknotet (Schritt 3) und verteilt sich im Arbeitsraum. Es sieht dann ähnlich aus wie in Fig. 1.2, nur mit mehr Knoten (siehe Abb. 11.4 in [1]). Gleichzeitig wird die Positionierung verbessert (Schritt 6). Jeder Knoten positioniert den Endeffektor auf Zielpunkte, die in seiner „Nähe“ liegen. Insgesamt werden schon nach 6.000 Trainingsschritten sehr gute Positionierungsergebnisse erzielt, was vor allem an der Verwendung der Jakobimatrizen A_r liegt.

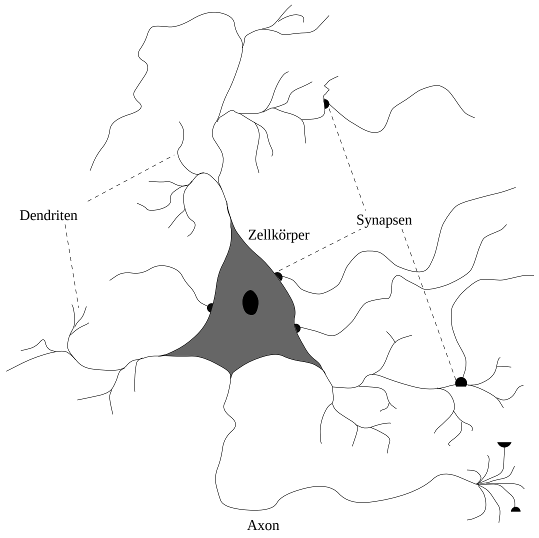
2. Selbstorganisierende künstliche Neuronale Netze - Kohonens Modell

Kohonen wollte 1982 mit der Entwicklung seines Modells ([2], S. 119-157) unter anderem die effiziente Speicherung von Daten erforschen. Er orientierte sich dabei, was den Aufbau seines Netzes betrifft, an einem biologischen Modell von Neuronen. Das Netz soll den Datenraum möglichst gut überdecken. Außerdem sollen, wie im biologischen Modell, „ähnliche Daten" an „benachbarten Stellen" gespeichert werden.

2.1 Abriß eines biologischen Modells eines Neurons

Ein Neuron enthält im wesentlichen drei Hauptstrukturen:

1. die *Dendriten* für die Eingabe,
2. den *Zellkörper* (Soma) für die Verarbeitung und
3. das *Axon* für die Ausgabe.



Die Dendriten sind sehr stark verzweigt und im Radius von ca. 400 μm um ein Neuron angeordnet. Sie summieren die elektrischen Signale anderer Neuronen und leiten diese an den Zellkörper weiter.

Überschreitet die Summe aller Signale einen *Schwellenwert*, so wird vom Zellkörper ein kurzzeitiger elektrischer Nadelimpuls über das Axon ausgegeben.

Das Axon, dessen Länge von weniger als einem Millimeter bis zu mehreren Metern variiert, leitet diesen Impuls weiter. An dem stark verzweigten Ende des Axons befinden sich *Synapsen*, die sich an den Dendriten und Zellkörpern anderer Neuronen anlagern. Die Übertragung des elektrischen Impulses findet durch chemische Botenstoffe (Neurotransmitter) statt. Diese bestimmen die *Synapsenstärke* und können je nach Beschaffenheit entweder *hemmend* oder *erregend* wirken.

2.2 Definition eines selbstorganisierenden künstlichen Neuronalen Netzes

Ein selbstorganisierendes künstliches Neuronales Netz in Kohonens Modell besteht aus einer Menge von Datenvektoren und einer endlichen Menge von Units (statt Neuronen, um die Distanz zur Biologie zu wahren), die jeweils mit einer Metrik ausgestattet sind. Durch diese Metriken ist der *Abstand* zwischen Units bzw. Datenvektoren bestimmt. Die induzierten Topologien definieren *Umgebungen* (auch *Nachbarschaften* genannt) von Units bzw. Datenvektoren. Die *Nachbarn* einer Unit r sind alle Units, die in einer bestimmten Umgebung von r liegen (analog für Datenvektoren). Jeder dem Netz präsentierte Datenvektor soll einer Unit zugeordnet werden. Es wird folgende Forderung der *Nachbarschaftserhaltung* gestellt:

> Die Daten sollen homöomorph auf die Units abgebildet werden, in dem Sinne, daß benachbarte Daten benachbarten Units zugeordnet werden.

Sei $X \subset \mathbb{R}^d$ die Menge der Daten und $x = (x_1, \ldots, x_d) \in X$. Das Netz habe d Eingangsfasern, und x_i bezeichne die i-te Faser. Sei y_r die Ausgabe (Potential des Axons) und s_r der Schwellenwert von Unit r, wobei $r \in \mathbb{R}^n$ der *Ortsvektor* der Unit ist. Der *Abstand zweier Units* ist durch den Abstand ihrer Ortsvektoren in $\mathbb{R}^n$ gegeben. Jede Unit ist mit allen Eingangsfasern verbunden, und w_{ri} bezeichne die Verbindungsstärke (Synapsenstärke) der Eingangsfaser i zu Unit r. Zur Berechnung der Ausgabe y_r einer Unit r schlägt Kohonen folgende Gleichung vor:

$$y_r(x) = \phi\left(\sum_{i=1}^{d} w_{ri} x_i + \sum_{q \neq r} g(r,q) y_q(x) - s_r\right) \qquad (1)$$

$$\text{mit } \phi : \mathbb{R} \to [0,1] \text{ und } \ \phi(x) \to 1 \text{ bzw. } 0 \ \text{ für } \ x \to \pm\infty.$$

Der erste Term summiert die Werte aller Eingangsfasern gewichtet mit ihrer Synapsenstärke. Durch den zweiten Term wird die Ausgabe der Nachbarn von Unit r mit einbezogen. Die Funktion $g(r,q)$ bestimmt dabei den Einfluß, den Unit q auf Unit r hat. Ist $g(r,q)$ für kleinen Abstand von r und q erregend ($g(r,q) > 0$) und für großen Abstand hemmend ($g(r,q) < 0$), dann wird die von g bewirkte Rückkopplung auch als *laterale Inhibition* oder *Umfeldhemmung* bezeichnet. Eine Mexican-Hat Funktion bewirkt zum Beispiel solch eine Rückkopplung. Jedem Signal $x \in X$ wird eine Unit r' mit *maximaler Ausgabe* zugeordnet.

Nun läßt sich zeigen: Die Funktion g kann so gewählt werden, daß die oben gestellte Forderung der Nachbarschaftserhaltung erfüllt wird ([1], S. 69-71).

Um zu gegebenem $x \in X$ die zugeordnete Unit r' mit maximaler Ausgabe zu bestimmen, muß das nichtlineare Gleichungssystem (1) numerisch gelöst werden. Kohonen vereinfacht die Suche nach dem *Erregungszentrum* r', indem dieses nur noch aufgrund des präsentierten Datenvektors x berechnet wird, ohne die Nachbarunits einzubeziehen (sei ohne Einschränkung $s_r := 0$):

$$\sum_{i=1}^{d} w_{r'i} x_i = \max_r \sum_{i=1}^{d} w_{ri} x_i. \qquad (2)$$

Mit den durch Normierung zu erreichenden Voraussetzungen

$$\|w_r\|_2 = 1 \ \ \forall\, r \quad \text{und} \quad \|x\|_2 = 1 \ \ \forall\, x \in X \quad \text{gilt:}$$

$$\|w_r - x\|_2^2 = \|w_r\|^2 - 2\langle w_r, x\rangle + \|x\|^2 = 2 - 2\sum_{i=1}^{d} w_{ri} x_i.$$

$$\text{Damit folgt} \quad (2) \iff \|w_{r'} - x\|_2^2 = \min_r \|w_r - x\|_2^2.$$

Das Erregungszentrum r' ist demnach eine Unit, deren Verbindungsstärke zu den Eingangsfasern den kleinsten Abstand zum präsentierten Datenvektor x hat. r' wird oft auch als *Gewinnerunit* bezeichnet. $w_r \in \mathbb{R}^d$ kann als *Wert* von Unit r interpretiert werden, die Ausgabe y_r wird nicht mehr betrachtet. Der Wert einer Unit muß von ihrem Ortsvektor $r \in \mathbb{R}^n$ unterschieden werden.

Künstliche Neuronale Netze sind in der Lage, das Lösen von gestellten Aufgaben zu „erlernen“. Unter Lernen versteht man, die sich schrittweise nach gewissen *Lernregeln* verändernde Zuordnung von Eingangssignalen zu einem Ausgabewert. Die Zuordnung soll dabei so verändert werden, daß im Laufe des *Lernverfahrens* die gestellte Aufgabe immer „besser“ gelöst wird.

Der Ausgabewert eines selbstorganisierenden künstlichen Neuronalen Netzes ist die dem Signal x zugeordnete Unit r'. Um die Datenvektoren möglich gut zu überdecken, schlägt Kohonen folgende *Lernregel* vor:

$$\forall\ r: \quad \Delta w_r = \epsilon \Lambda(r', r)(x - w_r)$$

$\epsilon \in [0, 1]$ ist ein *Lernschrittparameter*. Die Nachbarn einer Unit werden bei der Berechnung des Erregungszentrums r' im Gegensatz zu (1) nicht berücksichtigt. Deswegen führt Kohonen die *Nachbarschaftsfunktion* $\Lambda(r', r) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$ ein. Diese hat folgende Eigenschaft:

$$\Lambda(r', r_1) > \Lambda(r', r_2), \quad \text{falls} \quad \|r' - r_1\| < \|r' - r_2\|. \tag{3}$$

Je weiter r von der Gewinnerunit r' entfernt ist, desto kleiner wird $\Lambda(r', r)$. Der Abstand des Vektors w_r zum Datenvektor x wird durch einen Lernschritt verkleinert, und zwar in Abhängigkeit von dem Abstand von r und r'.

Eine typische Wahl für eine Nachbarschaftsfunktion ist die Glockenkurve

$$\Lambda(r', r) = \exp\left(-\frac{\|r' - r\|^2}{2\sigma^2}\right).$$

Mit dem Parameter σ kann die Breite der „Glocke“ und somit die *Größe der Nachbarschaft* um die Gewinnerunit r' gesteuert werden.

2.3 Der Algorithmus für ein selbstorganisierendes Netz

1. Wähle $w_{ri} \in [0, 1]$ zufällig, σ und ϵ „groß“.

2. Wähle ein sensorisches Signal $x \in X$ unter Beachtung einer Wahrscheinlichkeitsverteilung P auf X.

3. Bestimme das Erregungszentrum r' gemäß

$$\|x - w_{r'}\|_2 = \min_r \|x - w_r\|_2. \tag{4}$$

4. Setze
$$\forall\, r: \quad w_r^{neu} := w_r^{alt} + \epsilon\Lambda(r', r)(x - w_r^{alt}).$$

5. Reduziere ϵ und σ. Stopp, falls ϵ „klein genug“, sonst fahre mit Schritt 2 fort.

Zunächst einige Bemerkungen zum Algorithmus:

Zu 1: σ ist im wesentlichen die Größe der Nachbarschaft um r', für die $\Lambda(r', r)$ in Schritt 4 so groß ist, daß der zweite Term nicht völlig verschwindet. σ „groß“ bedeutet $\sigma \approx \max_{r_1, r_2} \|r_1 - r_2\|$, dem größten Abstand zwischen zwei Units. ϵ „groß“ heißt $\epsilon \approx 1$.

Zu 2: Gewöhnlich ist P die Gleichverteilung auf X.

Zu 3: Obwohl weder w_r noch x normiert sind, wird (4) zur Bestimmung von r' benutzt. (4) ist ohne Normierung nicht mehr äquivalent zur Maximumbedingung (2).

Zu 4: Es müssen nicht alle w_r neu gesetzt werden, sondern nur solche mit $\Lambda(r', r) > \lambda$. λ ist dabei ein kleiner Wert, für den Schritt 4 noch Sinn macht.

Zu 5: Bei der Reduzierung von ϵ und σ ist zu beachten, daß einerseits eine sehr langsame Reduzierung die Gesamtlaufzeit stark erhöht, andererseits eine zu schnelle Reduzierung dem anfangs zufällig gewählten Netz kaum eine Möglichkeit läßt, sich auszubreiten und die Topologie von X zu finden.

Des weiteren muß das Abbruchkriterium ϵ „klein genug“ und damit die Gesamtanzahl der Iterationen schon zu Beginn des Algorithmus gewählt werden. Natürlicherweise sollte der Algorithmus terminieren, falls sich das Netz dem Datenraum X „gut genug“ angepaßt hat.

2.4 Unitabstand und Nachbarschaftsfunktion

Der Ortsvektor einer Unit sei im folgenden durch $r \in \mathbb{N}_0^n$ gegeben. Den Abstand zwischen zwei Units (*Unitabstand*) r und r' definiere ich durch

$$\|r - r'\|_1 = \sum_{i=1}^{n} |r_i - r'_i| \in \mathbb{N}_0. \tag{5}$$

$\|.\|_1$ hat wegen ihrer ganzzahligen Werte zusammen mit der unten gewählten Nachbarschaftsfunktion Λ, im Gegensatz zur oft gewählten $\|.\|_2$ den Vorteil, daß mit dem Parameter σ die Größe der Nachbarschaft um die Gewinnerunit r' bei Handsteuerung (siehe 2.7) exakt gewählt werden kann. Alle Units, die in der σ-Nachbarschaft von r' liegen, werden in Schritt 4 des Algorithmus verändert.

Zur Veranschaulichung des Abstands zweier Units kann man sich zum Beispiel ein zweidimensionales Netz mit 256 Units als 16×16 Matrix vorstellen. Der Wert einer Unit wird durch einen Matrixeintrag aus $\mathbb{R}^d$ repräsentiert, der zweidimensionale Ortsvektor ist durch den Zeilen- und Spaltenindex des Wertes bestimmt. Die direkten Nachbarn von Unit (i, j) stehen eine Spalte weiter rechts bzw. links $(i, j \pm 1)$ und eine Zeile weiter unten bzw. oben $(i \pm 1, j)$, ohne dabei den Rand zu überschreiten.

Für die Nachbarschaftsfunktion Λ wähle ich eine lineare Funktion, für die Bedingung (3) offensichtlich erfüllt ist (siehe Fig. 2.1):

$$\Lambda(r', r) = \begin{cases} 1 - \frac{\|r'-r\|_1}{\sigma+1} & \text{falls } \|r' - r\|_1 \leq \sigma \\ 0 & \text{sonst} \end{cases}$$

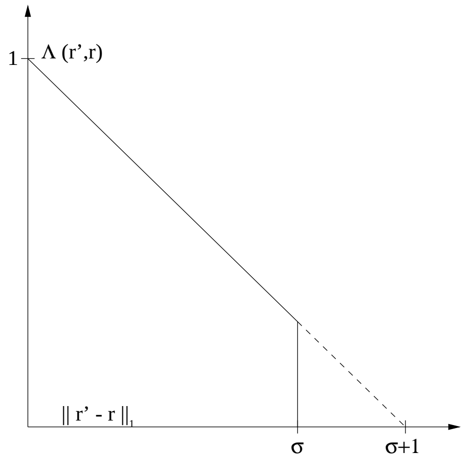


Fig. 2.1: die Nachbarschaftsfunktion Λ

Schritt 4 des Algorithmus wird für alle Units r ausgeführt mit $\Lambda(r', r) \neq 0$, also falls r noch in der σ-Nachbarschaft des Erregungszentrums r' liegt. $\sigma = 1$ bedeutet, daß $\Lambda(r', r) \neq 0$ nur für die unmittelbaren Nachbarn von r' gilt, für diese hat Λ den Wert $\frac{1}{2}$. Für $\sigma = max_{r_1,r_2}\|r_1 - r_2\|_1$ ist $\Lambda(r', r) \neq 0 \;\; \forall \; r$. In Schritt 4 werden dann die Werte aller Units in Abhängigkeit von ihrer Entfernung zu r' verändert.

Um ein Fehlermaß für das Netz zu erhalten, interpretiere ich den in (4) erhaltenen minimalen Abstand einer Unit zum präsentierten Datenvektor als *Fehler* des Netzes. Dieser wird über jeweils zehn Iterationsschritte gemittelt. Die zugehörige Kurve ist zusammen mit den Netzen zu sehen.

2.5 Beispiele

Alle abgebildeten Beispiele sind repräsentativ für den jeweiligen Netztyp und Datenraum, das heißt, es ist problemlos möglich, die abgebildeten Netze in ähnlicher Form mit der angegebenen Parametersteuerung zu reproduzieren.

Im folgenden werden zwei Standardbeispiele diskutiert. In beiden Fällen ist P die Gleichverteilung auf $X = [0, 1]^2$. Die Steuerung der Parameter ϵ und σ ist in 2.7 angegeben.

Die einzelnen Units sind in den Graphiken nicht direkt dargestellt, sondern die Werte benachbarter Units sind durch eine Linie verbunden. Für $n > 1$ sind die Werte der Units implizit durch sich kreuzende Linien gegeben.

Als erstes Beispiel (siehe S. 12) ist ein quadratisches Netz der Dimension $n = 2$ mit 256 Units zu sehen. Nach der Initialisierung sind die Units zufällig in ganz X verteilt. In den meisten mir bekannten Publikationen werden die Units in einer kleinen Umgebung um den Mittelpunkt von X verteilt. Wird der Parameter σ anfangs so groß wie der maximale Abstand zwischen zwei Units gewählt, dann ist diese einschränkende Wahl nicht notwendig. In diesem Fall werden, wie schon erläutert, in jedem Iterationsschritt die Werte aller Units in Schritt 4 des Algorithmus verändert. Diese Veränderung ist wegen der Nachbarschaftsfunktion Λ abhängig vom Abstand zur Gewinnerunit r'. Dadurch wird das durch die zufällige Anfangskonfiguration völlig verknotete Netz langsam entknotet.

Das Netz ist schon nach 40 Iterationen knotenfrei, dabei überdeckt es nur noch einen kleinen Teil von X. Da die Werte aller Units in jedem Iterationsschritt verändert werden, wird das gesamte Netz zu dem präsentierten Datenvektor x „geschoben“. Deswegen ist die Ausbreitung des Netzes bei großem σ klein. σ wird mit wachsender Iterationsanzahl verkleinert. Dadurch breitet sich das Netz immer stärker aus und nähert sich dem Rand des Quadrats. Nach 500 Schritten hat sich das Netz schon gut verteilt, jetzt geht es nur noch darum, möglichst dicht an den Rand heranzukommen.

Dieser kann nie erreicht werden, es wird immer ein Abstand bleiben. Der Abstand vom Rand wird, bei gleichverteilten Eingangswerten, mindestens halb so groß sein, wie der Abstand der äußersten Units zu den zweitäußersten. Die äußersten Units werden nämlich sowohl zum Rand (durch Daten, die zwischen dem Rand und ihnen liegen), als auch zu den zweitäußersten Units (durch Daten, die weiter innen liegen) „gezogen“.

Das zweite Beispiel (siehe S. 13) zeigt ein eindimensionales Netz mit ebenfalls 256 Units. Direkt nach der Initialisierung sieht es wie das vorige aus. Nach 40 Iterationen ist ein knotenfreies Netz zu erkennen. Nach 80 Schritten hat es sich im ganzen Raum ausgebreitet, danach beginnt es sich zu falten. Die Wellenlänge der Faltung ist durch die Form des Raumes bzw. des Netzes und σ bestimmt. Eine ausführliche Diskussion mit der Bestimmung von kritischen Werten, ab denen sich ein Netz faltet, ist in [1] S. 275-290 nachzulesen.

2.6 Untersuchung der Fehlerfunktion

Bei der Betrachtung der zugehörigen Fehlerkurven fällt zunächst deren sehr ähnlicher Verlauf auf. Sie starten mit relativ kleinen Werten (zufällige Anfangsverteilung im ganzen Raum), steigen stark an (geringe Überdeckung des Raumes) und fallen bis 100 Iterationen wieder schnell ab (beginnende Ausbreitung im Raum). Die Ausbreitung der Netze im Datenraum X ist demnach direkt an den zugehörigen Fehlerkurven abzulesen, was auch aus der Definition des Fehlers (4) folgt. Bis 1.500 Iterationen fallen die Kurven dann nur noch langsam, und sie haben ungefähr den gleichen Schlußwert (ca. 0.025).

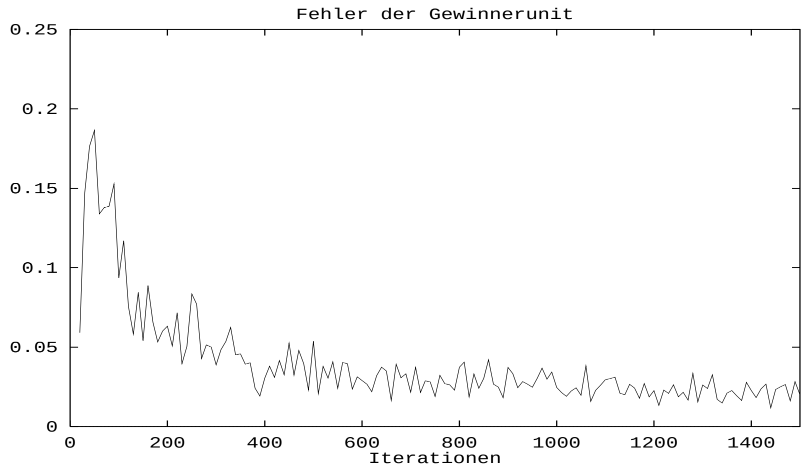
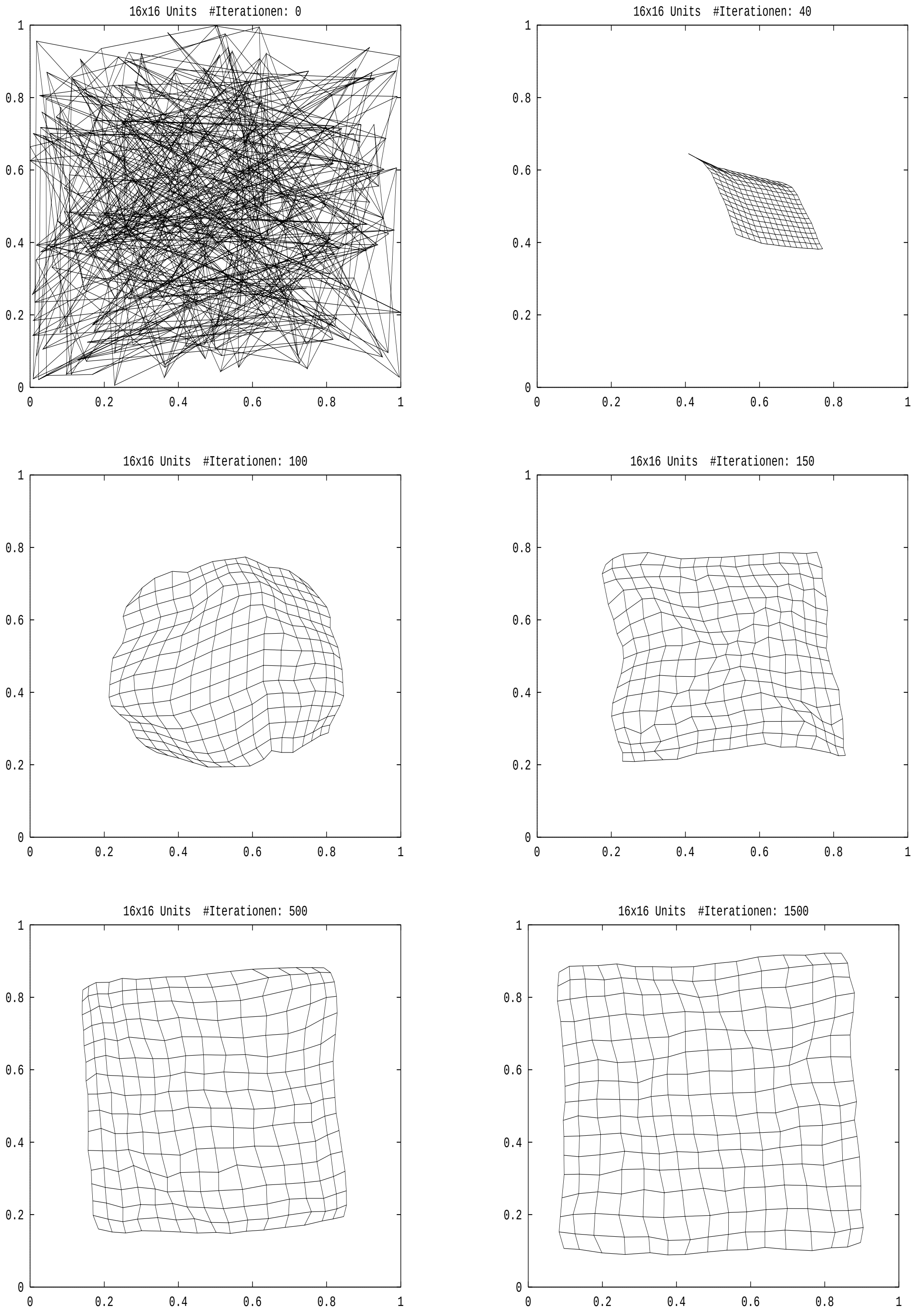
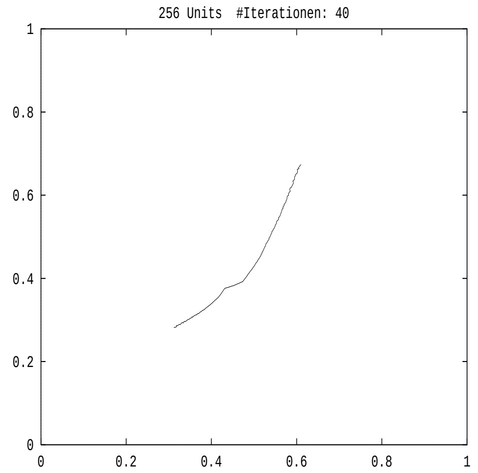
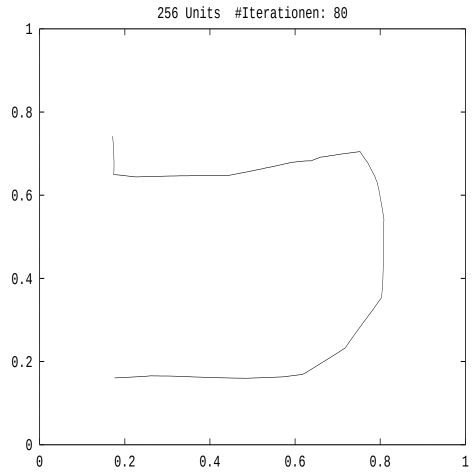


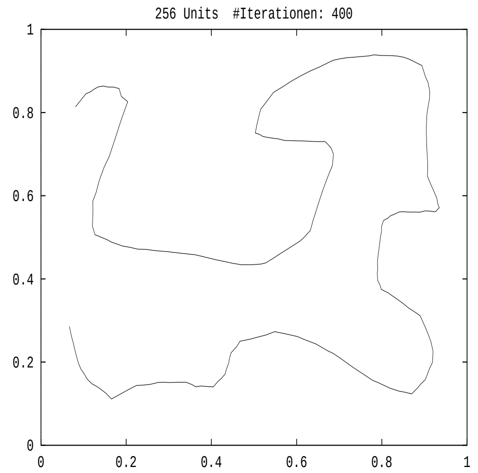
Fig. 2.2: Fehlerkurve zum 16 × 16 Netz (Beispiel 1)

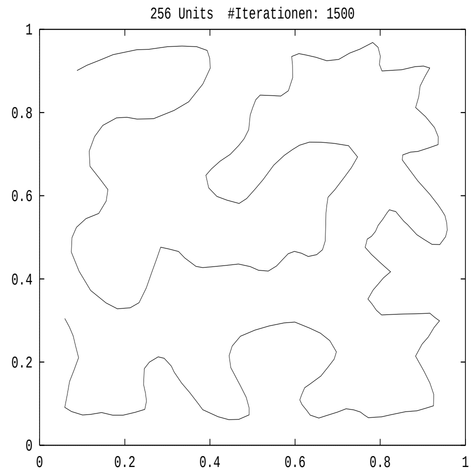
Beispiel 1

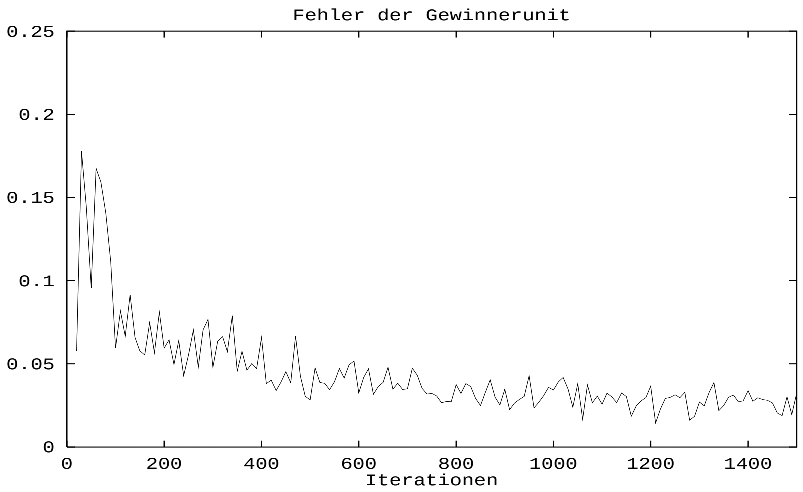

16x16 Units #Iterationen: 0
16x16 Units #Iterationen: 40
16x16 Units #Iterationen: 100
16x16 Units #Iterationen: 150
16x16 Units #Iterationen: 500
16x16 Units #Iterationen: 1500


256 Units #Iterationen: 40
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


256 Units #Iterationen: 80
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


256 Units #Iterationen: 400
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


256 Units #Iterationen: 1500
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Fehler der Gewinnerunit
0.25
0.2
0.15
0.1
0.05
0
0
200
400
600
800
1000
1200
1400
Iterationen

Der Schlußwert des Fehlers für das erste Beispiel läßt sich, falls die Units völlig gleichmäßig und unter Beachtung der Nachbarschaft verteilt sind, nach oben abschätzen:

$$\forall\, x \in X: \quad \|x - w_{r'}\|_2 \le \frac{1}{2}\sqrt{\frac{1}{17} + \frac{1}{17}} = \frac{1}{\sqrt{34}} \approx 0.171. \tag{6}$$

Es gibt vertikal 16 Units; der freie Rand sorgt dafür, daß der „vertikale" Abstand $\|w_{ij} - w_{ij+1}\|_2$ zwischen benachbarten Units $\frac{1}{17}$ beträgt. Der Wurzelausdruck in (6) beschreibt den „diagonalen" Abstand, also $\|w_{ij} - w_{i+1j+1}\|_2$. Eine obere Abschätzung für den Fehler erhält man nun durch Halbieren. Nach unten ist der Fehler durch 0 beschränkt, da eine präsentierte Eingabe mit einem w_{ij} übereinstimmen kann. Der durchschnittliche Fehler wird deutlich kleiner sein, als das arithmetische Mittel 0.086, da es „nur" $17 * 17 = 289$ Punkte in X gibt, für die in (6) Gleichheit gilt.

Dies zeigt auch die Fehlerkurve des ersten Beispiels. Dort ist der mittlere Fehler noch etwas kleiner, da die Units nicht völlig gleichmäßig verteilt sind und deswegen dichter liegen, als oben angenommen.

Der Datenraum X wird in beiden Beispielen gleichgut überdeckt, in dem Sinne, daß der Abstand eines Datenvektors zu einer Unit möglichst klein wird. Das ist nicht erstaunlich, wenn man bedenkt, daß in beiden Fällen 256 Units „gleichmäßig" verteilt werden. Es ist allerdings erstaunlich, daß das eindimensionale Netz keine Überschneidungen aufweist.

Die Steuerung der Parameter ϵ und σ (siehe 2.7) wurde in den beiden nächsten Beispielen mit der Hand durchgeführt, indem interaktiv abhängig vom Entwicklungsstand des Netzes neue Werte eingegeben wurden. Für Fig. 2.3 ist die Steuerung ähnlich wie in 2.7 für das zweidimensionale Netz, wobei die kleinere Netzgröße zu beachten ist. Ein Knoten (Fig. 2.4) kommt dadurch zustande, daß σ von Anfang an sehr klein gewählt wird ($\sigma \le 3$). Allerdings hat das Netz selbst bei dieser „schlechten" Steuerung nicht immer einen Knoten. Die genauen Werte für die Handsteuerung sind nicht angegeben, da mit diesen nur manchmal ein Knoten entsteht. Um die Fehlerkurven besser vergleichen zu können, wurde auch mit Handsteuerung bis 1.500 Iterationen gerechnet, obwohl beide Netze schon nach 800 Schritten ähnlich aussehen, wie jetzt abgebildet.

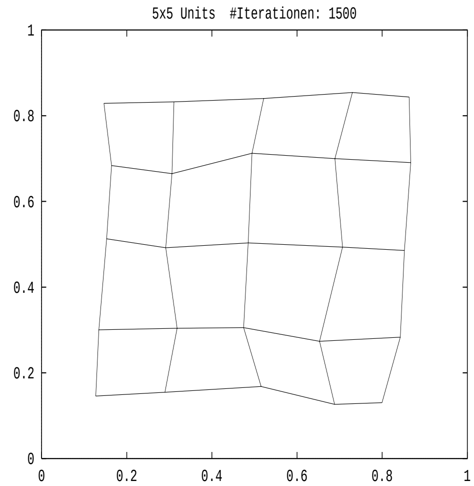


Fig. 2.3: ein Netz mit wenigen Units

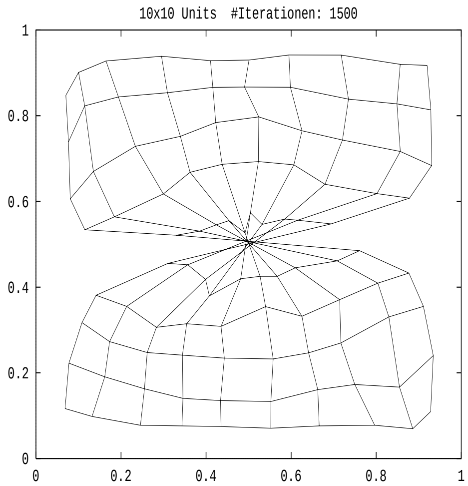


Fig. 2.4: ein verknotetes Netz

Der Fehler ist nicht von der Dimension des Netzes abhängig, sondern, wie das 5×5 Netz (Fig. 2.3 und 2.5) zeigt, von der Gesamtanzahl der Units. Diese ist mit 25 um den Faktor 10 kleiner, als bei den beiden bisher betrachteten Netzen. Die Fehlerkurve verläuft insgesamt auf einem höheren Niveau und endet bei einem doppelt so hohen Wert. Das ist auch zu erwarten, da die Abstände zwischen den Units deutlich größer sind.

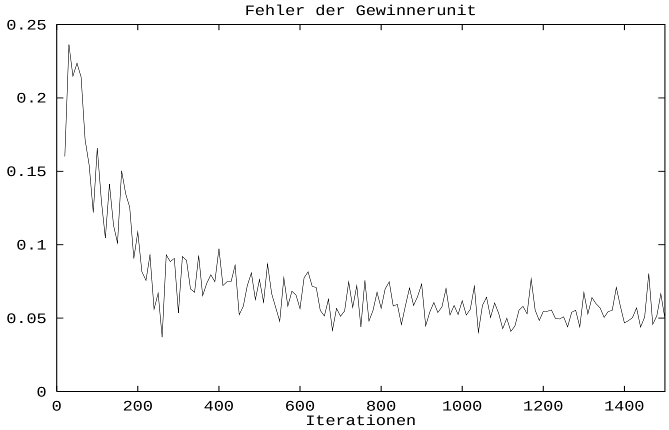


Fig. 2.5: Fehlerkurve zum 5x5 Netz (Fig. 2.3)

Es ist nicht möglich, mit Hilfe dieser Fehlerfunktion zu erkennen, ob ein Netz einen *topologischen Defekt*, zum Beispiel einen Knoten (Fig. 2.4) hat. Ein Netz mit solch einem Defekt unterscheidet sich in den Werten der Gleichung (4) praktisch nicht von einem Netz ohne Defekt, da es sich in X vergleichbar ausbreitet. Die zugehörige Fehlerkurve (Fig. 2.6) unterscheidet sich daher weder im Verlauf noch im Schlußwert wesentlich von denen der vorherigen Beispiele.

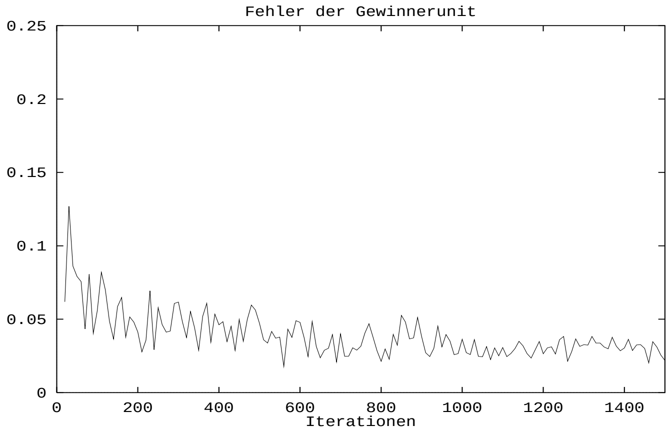


Fig. 2.6: Fehlerkurve zum verknoteten Netz (Fig. 2.4)

Für die beiden eben beschriebenen Probleme, die Untersuchung, ob Netz- und Datendimension übereinstimmen und das Erkennen von topologischen Defekten, wird in den folgenden Kapiteln eine Lösung beschrieben und genauer untersucht.

2.7 Steuerung der Parameter ϵ und σ

Die Anzahl von 1.500 Iterationsschritten ist im Vergleich mit Werten aus der Literatur sehr klein. Das liegt vor allem daran, daß dort die Parameter ϵ und σ sehr langsam verkleinert werden. Damit steigt zwangsläufig die Gesamtanzahl der Iterationen, teilweise (siehe [1] und [2]) auf 50.000 – 100.000. Diese große Anzahl ist sinnvoll, falls X eine endliche Menge ist. Dann muß die Iterationsanzahl in einem vernünftigen Verhältnis zur Mächtigkeit von X stehen. Allerdings ist die hohe Anzahl nicht notwendig, wenn X eine unendliche Menge ist, da nicht „alle“ Elemente von X präsentiert werden können.

Wird die Steuerung von ϵ und σ nicht vorher festgelegt und werden stattdessen abhängig vom Entwicklungsstand des Netzes interaktiv neue Werte eingegeben (Handsteuerung), dann ist eine Gesamtanzahl der Iterationen deutlich unter 1.000 problemlos zu erreichen. Dieses Verfahren bietet sich für alle Fälle an, in denen man sich das Netz ansehen kann, also falls die Dimension des Netzes und die des Datenraums kleiner als vier sind ($n, d < 4$). Die weitaus meisten der bisher in der Literatur diskutierten Fälle sind von diesem Typ.

In keiner der Publikationen [2], [3] und [4] sind zu einem gerechneten Beispiel beide zu wählenden Parameter $\epsilon(t)$ und $\sigma(t)$ angegeben. Lediglich in [1] wurden bei einem Beispiel beide Funktionen angegeben.

Sei $D = \max_{r_1,r_2} \|r_1 - r_2\|_1$ der größte Abstand zwischen zwei Units und N deren Gesamtanzahl. Dann haben sich durch viele gerechnete Beispiele für $N \geq 100$ folgende Funktionen $\epsilon(t)$ und $\sigma(t)$ als günstig erwiesen:

- für ein zweidimenionales Netz

#Iterationen	50	75	100	200	300	400	500	> 500
$\sigma(t)$	D	$\frac{3}{4}D$	$\frac{1}{2}D$	$\frac{2}{5}D$	$\frac{1}{3}D$	$\frac{1}{5}D$	$\frac{1}{7}D$	$\frac{1}{10}D$
$\epsilon(t)$	$\frac{1}{4}$			$\frac{1}{20}$				

- für ein eindimensionales Netz

#Iterationen	25	50	75	100	300	500	600	> 600
$\sigma(t)$	$\frac{2}{3}D$	$\frac{1}{2}D$	$\frac{1}{3}D$	$\frac{1}{5}D$	$\frac{1}{10}D$	$\frac{1}{20}D$	$\frac{1}{35}D$	$\frac{1}{75}D$
$\epsilon(t)$	$\frac{1}{3}$					$\frac{1}{7}$		

In Fig. 2.7 ist der Verlauf der Funktionen $\sigma(t)/D$ für ein ein- bzw. zweidimensionales Netz graphisch dargestellt. Die unterschiedlichen Schlußwerte sind auf das für die beiden Dimensionen stark unterschiedliche D zurückzuführen. So gilt für die eben besprochenen Beispiele mit 256 Units $D_{n=2} = 32$ und $D_{n=1} = 256$, oder allgemein $D = nm$ für ein $m \times \ldots \times m$ Netz der Dimension n. Für den Schlußwert von σ bedeutet das: $\sigma_{n=2} = 3.2$ und $\sigma_{n=1} = 3.4$.

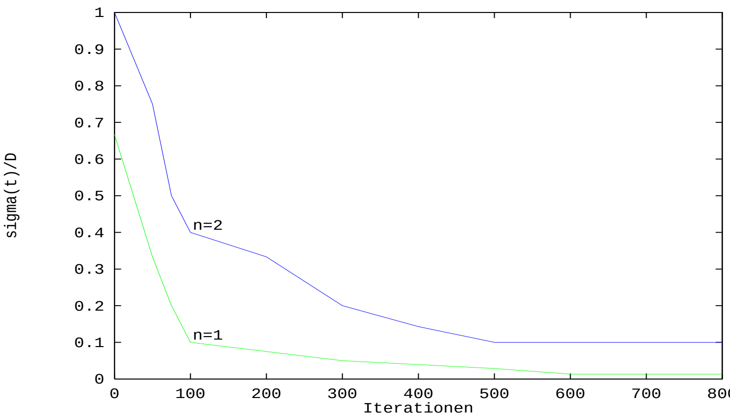


Fig. 2.7: Reduzierung des Parameters σ

3. Nachbarschaftserhaltung

In diesem Kapitel geht es darum, eine Lösung für die vorhin beschriebenen Probleme vorzustellen. Warum ist es überhaupt nützlich, topologische Defekte zu erkennen? Am Anfang von Kapitel 2 wurde die Forderung aufgestellt, daß alle Daten homöomorph auf die Units abgebildet werden sollen. Bei einem verknoteten Netz (Fig. 2.4) ist die Homöomorphie verletzt. Wie zu sehen ist, gibt es Units, deren *Unitabstand* (5) groß ist, deren *Werte* aber dicht beieinanderliegen. Nachbarschaft der Daten wurde nicht auf Nachbarschaft der Units übertragen, oder kurz: Die Nachbarschaft wurde nicht erhalten. Da es bisher nicht möglich ist, einen topologischen Defekt *automatisch* während des Algorithmus zu erkennen, bleibt nur die Möglichkeit, hinterher einen Defekt zu entdecken.

Ist die Dimension des Netzes kleiner als die des Datenraums, dann kommt es zur Faltung des Netzes. In diesem Fall (Seite 13) ist, wie bei dem verknoteten Netz, die Homöomorphieforderung nicht erfüllt, da es Units gibt, deren Abstand groß ist, deren Werte aber nahe beieinanderliegen.

Ist im umgekehrten Fall die Dimension des Netzes größer als die des Datenraums, dann kommt es ebenfalls zu einer Verletzung der Homöomorphie, wie Fig. 3.1 zeigt. Der Übersichtlichkeit wegen wurde die Vernetzung nur zweidimensional eingezeichnet; „übereinanderliegende“ Units der einzelnen „Schichten“ müßten ebenfalls noch verbunden werden (wie in Fig. 1.2). Ein voll verbundenes Netz ist auch in Fig. 8a von [5] zu sehen.

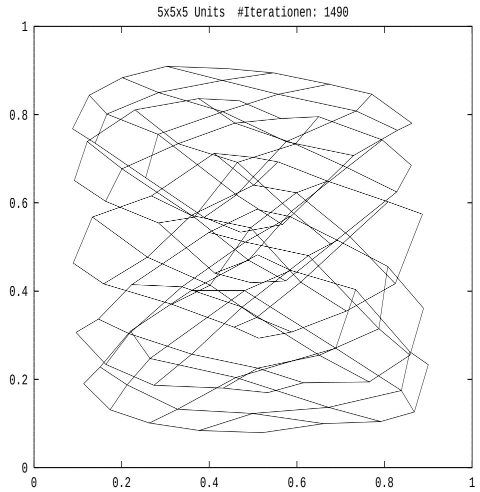


Fig. 3.1: ein dreidimensionales Netz

Sollen n-dimensionale Daten mit einem Kohonennetz repräsentiert werden, so wird man dieses zunächst mit einem n-dimensionalen Netz versuchen. Nun kann es aber sein, daß die Daten in einem im allgemeinen nichtlinearen Unterraum mit kleinerer Dimension liegen. Die zugehörige Abbildung vom Datenraum zum Netz ist dann nicht homöomorph. Könnte man diesen Effekt erkennen, dann gäbe es eine Möglichkeit, die „wirkliche“ Dimension des Datenraums herauszufinden.

Als ein Lösungsweg wird im folgenden der Ansatz aus [5] vorgestellt. Dieser baut wesentlich auf dem in [6] beschriebenen Topographischen Produkt auf.

3.1 Quantifizierung der Nachbarschaftserhaltung

Auch auf dem letzten Bild auf Seite 12 sind kleine Verletzungen der Nachbarschaftserhaltung zu erkennen. Diese treten auf, da das Netz nicht völlig gleichmäßig angeordnet ist. Deswegen ist es notwendig, die auftretenden Fehler bei der Nachbarschaftserhaltung zu quantifizieren.

3.1.1 Definition der nächsten Nachbarn

Sei X wieder der Datenraum, $Y := \{r \in I\!N_0^n |\ r$ bezeichnet eine Unit$\}$ die Menge aller Units und $r \in Y$ fest gewählt. Eine Folge der Ortsvektoren aller Units außer r, die monoton wachsend in dem Unit- bzw. Werteabstand zu r ist, wird als *Folge der Unitnachbarn U(r)* bzw. *Folge der Wertenachbarn W(r)* bezeichnet:

$$U(r) = (n_1^Y(r), n_2^Y(r), \ldots) : \quad i < j \implies \|r - n_i^Y(r)\|_1 \leq \|r - n_j^Y(r)\|_1$$
$$W(r) = (n_1^X(r), n_2^X(r), \ldots) : \quad i < j \implies \|w_r - w_{n_i^X(r)}\|_2 \leq \|w_r - w_{n_j^X(r)}\|_2$$

mit $n_i^Y(r), n_i^X(r) \in Y \backslash \{r\}$ und $i \neq j \implies n_i^Y(r) \neq n_j^Y(r) \ \wedge \ n_i^X(r) \neq n_j^X(r)$.
$n_k^Y(r)$ wird als *k-ter Unitnachbar* und $n_k^X(r)$ als *k-ter Wertenachbar* von r bezeichnet.

Da die Ortsvektoren ganzzahlig sind, ist die Folge der Unitnachbarn selbst bei einem eindimensionalen Netz nicht eindeutig bestimmt, unabhängig davon, ob der Unitabstand mit $\|.\|_1$ oder mit $\|.\|_2$, wie es die meisten Autoren tun, definiert wird. Bezeichne n wieder die Dimension des Netzes. Dann ist die Anzahl m der Unitnachbarn mit kleinstem Abstand durch $n \leq m \leq 2n$ bestimmt. Der Wert von m ist von der Lage der betrachteten Unit abhängig. Units am Rand des Netzes haben weniger direkte Unitnachbarn, als Units, die nicht am Rand liegen. Die Autoren von [5] umgehen das Problem der nicht eindeutig bestimmten Folge der Unitnachbarn, indem sie die Reihenfolge der Unitnachbarn mit gleichem Abstand zufällig bestimmen ([7]). Im nächsten Kapitel werden die Auswirkungen untersucht, wenn die Reihenfolge der Unitnachbarn mit gleichem Abstand auf andere Weise bestimmt wird.

Es kann auch Wertenachbarn geben, die denselben Werteabstand zu Unit r haben. Allerdings tritt dieser Fall wesentlich seltener auf, da die w_r kontinuierliche Werte annehmen.

3.1.2 Ein Maß für die Nachbarschaftserhaltung

Um die auftretenden Fehler der Nachbarschaftserhaltung zu messen, werden folgende Quotienten definiert:

$$\begin{aligned} \text{für den Werteabstand: } Q_X(r,k) &:= \frac{\|w_r - w_{n_k^Y(r)}\|_2}{\|w_r - w_{n_k^X(r)}\|_2} \\ \text{und für den Unitabstand: } Q_Y(r,k) &:= \frac{\|r - n_k^Y(r)\|_1}{\|r - n_k^X(r)\|_1}. \end{aligned} \tag{7}$$

Q_Y ist wohldefiniert, da die Ortsvektoren der Units verschieden sind und somit der Nenner ungleich null ist. Für Q_X gilt dasselbe, wenn angenommen wird, daß keine zwei Werte übereinstimmen. Damit gilt $Q_Y, Q_X > 0$.

Zur Veranschaulichung der Definitionen folgt in Fig. 3.2 ein stark vereinfachtes Beispiel. Dort ist ein eindimensionales Netz mit sieben Units zu sehen. Die Werte der Units sind hier als Punkte dargestellt, die an den Punkten stehenden Zahlen sind die (eindimensionalen) Ortsvektoren der Units. Die Unit- und Wertenachbarn der hier betrachteten Unit $r = 4$ sind Tabelle 3.1 zu entnehmen. Dabei ist zu beachten, daß die erste Zeile der Tabelle nicht eindeutig bestimmt ist. Die 1./2., 3./4. und 5./6. Unitnachbarn von Unit 4 können jeweils vertauscht werden, wodurch sich auch die Werte von Q_X verändern. Auf Q_Y hat eine Vertauschung keinerlei Auswirkungen, da zwei n_k^Y nur vertauscht werden, falls ihr Unitabstand zur betrachteten Unit gleich ist.

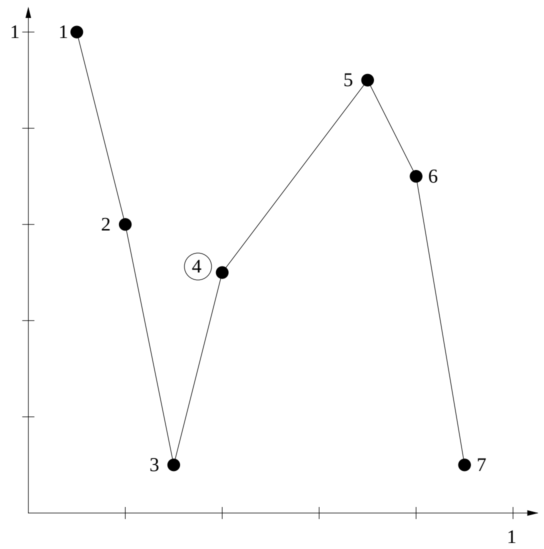


Fig. 3.2: Beispiel zur Nachbarschaftsdefinition

k	1	2	3	4	5	6
$n_k^Y(4)$	3	5	2	6	7	1
$n_k^X(4)$	2	3	6	5	1	7
$\lVert w_4 - w_{n_k^X(4)} \rVert_2$	$\frac{\sqrt{5}}{10}$	$\frac{\sqrt{17}}{10}$	$\frac{\sqrt{20}}{10}$	$\frac{\sqrt{25}}{10}$	$\frac{\sqrt{34}}{10}$	$\frac{\sqrt{41}}{10}$
$Q_X(4,k)$	$\sqrt{\frac{17}{5}}$	$\sqrt{\frac{25}{17}}$	$\frac{1}{2}$	$\sqrt{\frac{4}{5}}$	$\sqrt{\frac{41}{34}}$	$\sqrt{\frac{34}{41}}$
$Q_Y(4,k)$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{2}{2}$	$\frac{2}{1}$	$\frac{3}{3}$	$\frac{3}{3}$

Tab. 3.1: Tabelle zu Fig. 3.2

Unmittelbar aus der Definition der Quotienten folgt:

$$\begin{array}{ll} & \text{die k-ten Unit- und k-ten Wertenachbarn} \\ & \text{von } r \text{ stimmen überein.} \\ \Longleftrightarrow & n_k^X(r) = n_k^Y(r) \\ \Longrightarrow & Q_X(r,k) = 1 \text{ und } Q_Y(r,k) = 1 \end{array}$$

In [5] wird zusätzlich behauptet: $Q_X(r,k) = 1$ und $Q_Y(r,k) = 1 \Longrightarrow n_k^X(r) = n_k^Y(r)$. Diese Folgerung ist falsch, wie das Gegenbeispiel in Fig. 3.3 zeigt:

Dargestellt ist ein Ausschnitt eines eindimensionalen Netzes mit einer Unit $r \in I\!N$ und ihren beiden Unitnachbarn $r-1$ und $r+1$. Die Werte der drei Units sind als Punkte dargestellt. Ich nehme an, daß der Werteabstand von $r-1$ und $r+1$ zu r gleich ist. Für $U(r)$ und $W(r)$ gelte:

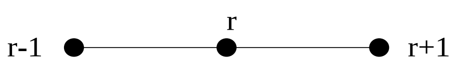


Fig. 3.3: Gegenbeispiel zur Folgerung in [5]

$$U(r) = (r-1, r+1, \ldots) \implies n_1^Y(r) = r-1 \ \wedge \ n_2^Y(r) = r+1$$
$$W(r) = (r+1, r-1, \ldots) \implies n_1^X(r) = r+1 \ \wedge \ n_2^X(r) = r-1.$$

Damit folgt:

$$Q_X(r,1) = \frac{\|w_r - w_{n_1^Y(r)}\|_2}{\|w_r - w_{n_1^X(r)}\|_2} = \frac{\|w_r - w_{n_2^X(r)}\|_2}{\|w_r - w_{n_1^X(r)}\|_2} = 1$$
$$\text{und} \quad Q_Y(r,1) = \frac{\|r - n_1^Y(r)\|_1}{\|r - n_1^X(r)\|_1} = \frac{\|r-(r-1)\|_1}{\|r-(r+1)\|_1} = 1.$$

Aber offensichtlich gilt: $n_1^X(r) = r+1 \neq r-1 = n_1^Y(r)$. □

Es wurde angenommen, daß die Werteabstände zweier Units zu einer Dritten gleich sind. Dieser Fall tritt, wie schon in 3.1.1 erwähnt, nur selten ein, da die w_r kontinuierliche Werte annehmen.

Die eben definierten Quotienten sind als Maß für die Nachbarschaftserhaltung zu empfindlich. Wenn die Elemente in X nicht gleichmäßig verteilt sind, sondern an einigen Stellen dichter liegen, dann konzentrieren sich die Units ebenfalls an diesen Stellen ([1], S. 80). Das Netz wird dort „engmaschiger" und damit die „Auflösung" höher. Die Quotienten zeigen schon bei diesem Fall durch von 1 abweichende Werte eine nicht vorhandene Nachbarschaftsverletzung an. Nicht gleichmäßig verteilte Signale treten bei fast allen praktischen Anwendungen auf.

Fig. 3.4 zeigt exemplarisch solch eine Situation. Es liegt wieder ein eindimensionales Netz vor, diesmal mit fünf Units. Der Datenraum X ist nicht mit abgebildet. Für die Nachbarschaften von Unit 3 gelte:

$$\begin{aligned} U(3) &= (4,2,5,1) \text{ und} \\ W(3) &= (4,5,2,1). \end{aligned}$$

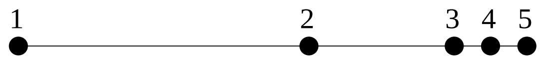


Fig. 3.4: die Werte der Units in X

$\implies Q_X(3,1) = Q_Y(3,1) = 1$, aber

$$Q_X(3,2) = \frac{\|w_3 - w_2\|_2}{\|w_3 - w_5\|_2} > 1 \quad \text{und} \quad Q_Y(3,2) = \frac{|3-2|}{|3-5|} < 1$$

In [5] wird darum vorgeschlagen, über die Quotienten zu mitteln. Das führt zu folgenden Definitionen:

$$P_X(r,k) := \left(\prod_{l=1}^{k} Q_X(r,l)\right)^{\frac{1}{k}} \qquad P_Y(r,k) := \left(\prod_{l=1}^{k} Q_Y(r,l)\right)^{\frac{1}{k}} \tag{8}$$

Das geometrische Mittel wird nicht über die Units, sondern über den Abstand zu Unit r gebildet. Durch diese Mittelung fallen einzelne Abweichungen nicht so sehr ins Gewicht. Es stellt sich heraus, daß sich der durch ein ungleichmäßig verteiltes Netz hervorgerufene Fehler der Nachbarschaftserhaltung in den Produkten (8) sehr viel weniger niederschlägt, als der Fehler eines gefalteten Netzes.

Für Fig. 3.4 ergeben diese Definitionen:

$$P_X(3,1) = P_Y(3,1) = 1, \;\; P_X(3,2) > 1, \;\; P_Y(3,2) < 1 \text{ und}$$
$$P_X(3,3) = \left(\frac{\|w_3 - w_4\|_2}{\|w_3 - w_4\|_2}\frac{\|w_3 - w_2\|_2}{\|w_3 - w_5\|_2}\frac{\|w_3 - w_5\|_2}{\|w_3 - w_2\|_2}\right)^{\frac{1}{3}} = 1, \;\; P_Y(3,3) = 1.$$

Für den Wertebereich der beiden Produkte gilt

Lemma 1: *Sind die Werte der Units paarweise verschieden, dann folgt*

$$\forall\, r \in Y, \; \forall\, k: \quad 0 < P_Y(r,k) \le 1 \le P_X(r,k) < \infty.$$

Beweis: Die erste und die letzte Ungleichung ergeben sich direkt aus der Definition.

$$(P_Y(r,k))^k = \prod_{l=1}^{k} Q_Y(r,l) = \frac{\|r - n_1^Y(r)\|_1}{\|r - n_1^X(r)\|_1} \cdots \frac{\|r - n_k^Y(r)\|_1}{\|r - n_k^X(r)\|_1}$$

Nach Definition der Nachbarschaften gilt sicherlich $Q_Y(r,1) \le 1$. Da die Abstände im Zähler nach Definition der Unitnachbarn n^Y monoton wachsen, ist in jedem Teilprodukt das Produkt der Zähler nicht größer, als das der Nenner, und somit gilt

$$\forall\, l: \quad Q_Y(r,1)\, Q_Y(r,2) \ldots Q_Y(r,l) \le 1 \quad \Longrightarrow \quad \forall\, k: \; P_Y(r,k) \le 1.$$

Da bei P_X die Abstände nach Definition im Nenner monoton wachsen, folgt analog $P_X(r,k) \ge 1$. $\square$

3.2 Dimensionserkennung

Die Werte der eben definierten Produkte lassen erkennen, ob die Dimension des Netzes und die des Raumes X übereinstimmen. Zur Motivation folgen zwei sehr idealisierte Beispiele mit $X = [0,1]^2$.

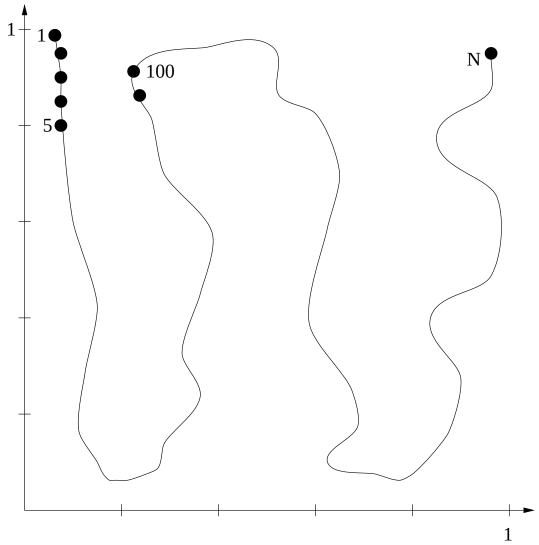


Fig. 3.5: „Dimension" von $X > dim(Y)$

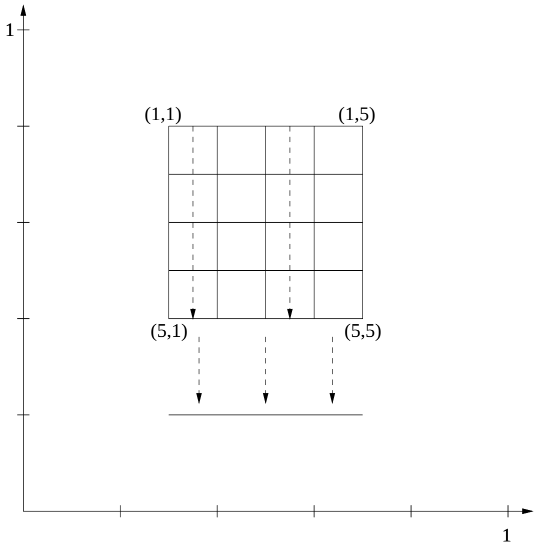


Fig. 3.6: „Dimension" von $X < dim(Y)$

Das erste Beispiel (Fig. 3.5) zeigt ein eindimensionales Netz mit N Units. Für die Nachbarschaften von Unit 1 gelte:

$$\begin{aligned} U(1) &= (2,3,4,5,6,7,\ldots) \text{ und} \\ W(1) &= (2,3,4,100,5,99,\ldots). \end{aligned}$$

$\Longrightarrow Q_X(1,k) = Q_Y(1,k) = 1$ für $k = 1,2,3,$ und

$$\begin{aligned} Q_X(1,4) &= \frac{\|w_1 - w_5\|_2}{\|w_1 - w_{100}\|_2} \approx 1 & Q_Y(1,4) &= \frac{|1-5|}{|1-100|} \ll 1 \\ Q_X(1,5) &= \frac{\|w_1 - w_6\|_2}{\|w_1 - w_5\|_2} \approx 1 & Q_Y(1,5) &= \frac{|1-6|}{|1-5|} \approx 1 \\ Q_X(1,6) &= \frac{\|w_1 - w_7\|_2}{\|w_1 - w_{99}\|_2} \approx 1 & Q_Y(1,6) &= \frac{|1-7|}{|1-99|} \ll 1 \end{aligned}$$

$\Longrightarrow P_X(1,k)P_Y(1,k) < 1$ für $k = 4,5,6$

Im zweiten Beispiel (Fig. 3.6) ist ein zweidimensionales Netz mit 25 Units zu sehen. Allerdings ist die Wahrscheinlichkeitsverteilung P nicht die Gleichverteilung auf X, sondern auf $[0.3, 0.7] \times \{0.2\}$. Alle Daten liegen also in der eingezeichneten Linie. Das Netz wird sich von der zufälligen Anfangskonfiguration in $[0,1]^2$ zu dieser Linie hin bewegen, das eingezeichnete Netz ist nur zur Veranschaulichung der Lage auf der Linie gedacht. Dort sind die einzelnen Units, allerdings wesentlich ungleichmäßiger und mit einigen Vertauschungen so angeordnet, wie durch die zwei Pfeile im Netz angedeutet. Das Netz wird im wesentlichen von oben auf die Linie zusammengedrückt, wobei sich die Units auf der Linie verteilen. Für Unit $(1,1)$ ergeben sich zum Beispiel folgende Nachbarschaften:

$$\begin{aligned} U(1,1) &= ((2,1),(1,2),(1,3),(2,2),(3,1),(1,4),\ldots) \text{ und} \\ W(1,1) &= ((2,1),(3,1),(4,1),(1,2),(5,1),(2,2),\ldots). \end{aligned}$$

$$\begin{aligned} \Longrightarrow \quad Q_X((1,1),1) &= \frac{\|w_{11} - w_{21}\|_2}{\|w_{11} - w_{21}\|_2} = 1 & Q_Y((1,1),1) &= \frac{\|(1,1)-(2,1)\|_1}{\|(1,1)-(2,1)\|_1} = 1 \\ Q_X((1,1),2) &= \frac{\|w_{11} - w_{12}\|_2}{\|w_{11} - w_{31}\|_2} > 1 & Q_Y((1,1),2) &= \frac{\|(1,1)-(1,2)\|_1}{\|(1,1)-(3,1)\|_1} = \frac{1}{2} \\ Q_X((1,1),3) &= \frac{\|w_{11} - w_{13}\|_2}{\|w_{11} - w_{41}\|_2} \gg 1 & Q_Y((1,1),3) &= \frac{\|(1,1)-(1,3)\|_1}{\|(1,1)-(4,1)\|_1} = \frac{2}{3} \end{aligned}$$

$\Longrightarrow P_X((1,1),k)P_Y((1,1),k) > 1$ für $k = 2,3$

Motiviert durch viele ähnliche Beispiele wird folgendes Produkt definiert:

$$P(r,k) := \sqrt{P_X(r,k)P_Y(r,k)}. \tag{9}$$

In [5] wird behauptet:

$$\begin{aligned} P > 1 &\implies \text{„Dimension" von } X < dim(Y) \text{ und} \\ P < 1 &\implies \text{„Dimension" von } X > dim(Y). \end{aligned} \tag{10}$$

Damit ergibt sich die Möglichkeit zu entscheiden, in welcher Relation die „Dimension" des Raumes X und die des Netzes stehen. In [5] wird jedoch keine schlüssige Begründung gegeben, warum die Folgerungen stimmen.

In der Anwendung zeigt sich jedoch, daß (10) ein sehr brauchbares Kriterium liefert. In [5] wird das Verfahren auf Sprachdaten angewendet. Dabei folgt das Ergebnis, daß ein dreidimensionales Netz besser geeignet ist, diese Daten zu repräsentieren, als ein zweidimensionales Netz. Bis zum Zeitpunkt der Veröffentlichung von [5] war man der Meinung, daß ein zweidimensionales Netz dafür besser geeignet ist.

Wie das Gehirn solche Daten speichert, läßt sich aber nicht beantworten. Schon das Modell von Kohonen hat mit dem Gehirn lediglich die Nachbarschaftseigenschaft gemein. Der Algorithmus hat allerdings, wie bereits erläutert, nicht mehr viel mit dem Modell zu tun.

3.3 Beispiele zur Dimensionserkennung

Zunächst werden Beispiel 1 und 2 aus 2.5 untersucht. Die zur Veranschaulichung gezeichneten Kurven von (8) und (9) sind nur noch von einer Variablen abhängig, dem Unitabstand k. Um dieses zu erreichen wurden die Werte von $P_X(r,k), P_Y(r,k)$ und $P(r,k)$ zum Unitabstand k jeweils über alle Units r arithmetisch gemittelt. Die so erhaltenen Funktionen werden mit $P_X(k), P_Y(k)$ und $P(k)$ bezeichnet. Das bedeutet zum Beispiel für P:

$$P(k) = \frac{1}{|Y|} \sum_{r \in Y} P(r,k).$$

Jede Abbildung wurde nach der Gesamtanzahl der Units skaliert. Wegen Lemma 1 liegt $P_X(k)$ oberhalb und $P_Y(k)$ unterhalb der 1. Nach (9) liegt $P(k)$ zwischen $P_X(k)$ und $P_Y(k)$.

Fig. 3.7 zeigt die Kurven zum ersten Beispiel, bei dem die Dimensionen übereinstimmen. Alle drei liegen nahe bei 1; insbesondere weicht $P(k)$ für $k > 30$ fast nicht von 1 ab. Wird die Anzahl der Iterationen sehr stark erhöht (siehe 2.7), dann liegen die Funktionen noch dichter an 1 (siehe [5], Fig. 7).

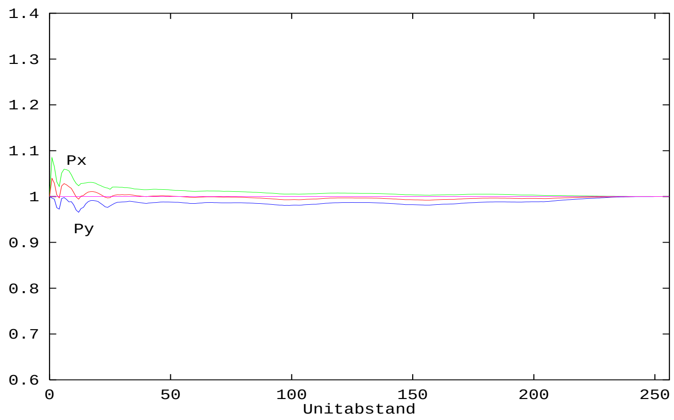


Fig. 3.7: Produktkurven zum 16 × 16 Netz (Beispiel 1, S. 12)

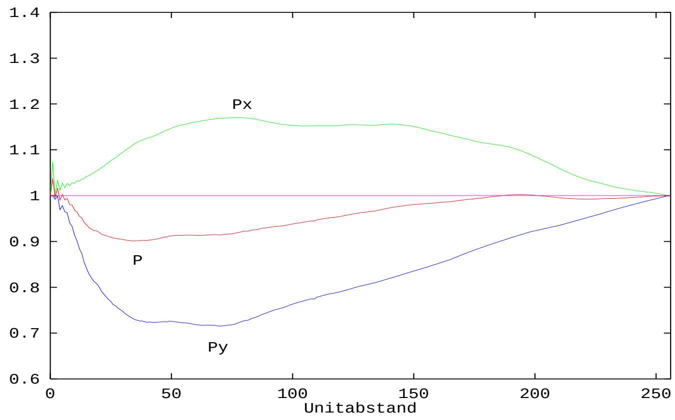


Fig. 3.8: Produktkurven zum 256 Netz (Beispiel 2, S. 13)

Ein deutlich anderer Verlauf der Kurven ist dagegen in Fig. 3.8 zu erkennen. Bei diesem zweiten Beispiel ist die Netzdimension kleiner als die Raumdimension. $P(k)$ liegt deutlich und durchgehend unterhalb von 1, wie in (10) behauptet.

Fig. 3.9 zeigt, daß bei zu großer Netzdimension $P(k)$ oberhalb von 1 liegt. Der Verlauf ist nicht so eindeutig wie in Fig. 3.8, aber im Vergleich zu Fig. 3.7 noch deutlich zu erkennen. Der unterschiedliche Verlauf der Kurven $P_X(k)$ und $P_Y(k)$ der letzten beiden Beispiele, langsamer bzw. schneller Anstieg und Abfall, tritt häufig, aber nicht immer bei zu kleiner bzw. großer Netzdimension auf. Es gibt Beispiele, bei denen der Verlauf der Kurven genau umgekehrt ist, insbesondere bei endlichem X.

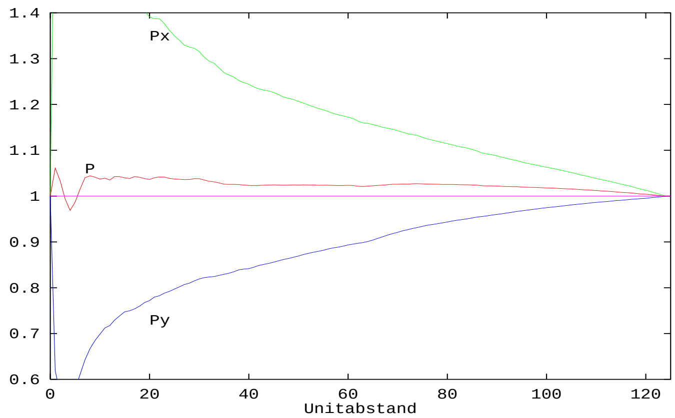


Fig. 3.9: Produktkurven zum $5 \times 5 \times 5$ Netz (Fig. 3.1)

Fig. 3.10 belegt, daß die Kurven nicht stark von der Gesamtanzahl der Units beeinflußt werden. Die Abweichungen von 1 sind größer, da das Netz sehr viel ungleichmäßiger verteilt ist und da das geometrische Mittel in den Produkten (8) und (9) für großen Unitabstand sehr viel stärker mittelt. Ein großer Abstand kommt aber wegen der geringen Anzahl der Units nicht zustande. Ein weiteres Beispiel dafür ist Fig. 5 in [5]: ein eindimensionales Netz mit 32 Units.

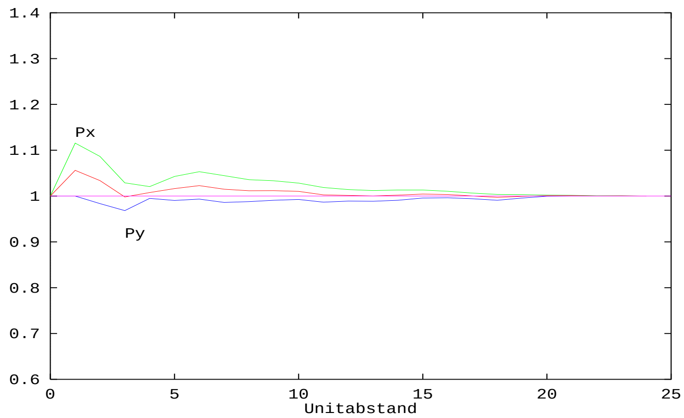


Fig. 3.10: Produktkurven zum 5×5 Netz (Fig. 2.3)

Hat das Netz einen topologischen Defekt, zum Beispiel einen Knoten, dann ist auch dieser Defekt an den Kurven abzulesen. Dazu zeigt Fig. 3.11 einen erstaunlichen Verlauf der Kurven. Obwohl $P_X(k)$ und $P_Y(k)$ sehr stark von 1 abweichen, gleichen sich ihre Werte bei der Mittelung zu $P(k)$ so aus, daß $P(k)$ nur für $k < 20$ von 1 abweicht. Der Verlauf von $P_X(k)$ und $P_Y(k)$ deutet auf eine Verletzung der Nachbarschaftserhaltung hin, während der Verlauf von $P(k)$ nahelegt, daß die Dimension des Netzes mit der des Raumes übereinstimmt; beides ist richtig. Ist der topologische Defekt nicht so regelmäßig wie in Fig. 2.4, dann verlaufen $P_X(k)$ und $P_Y(k)$ noch wesentlich weiter von 1 entfernt, $P(k)$ wird dagegen kaum beeinflußt.

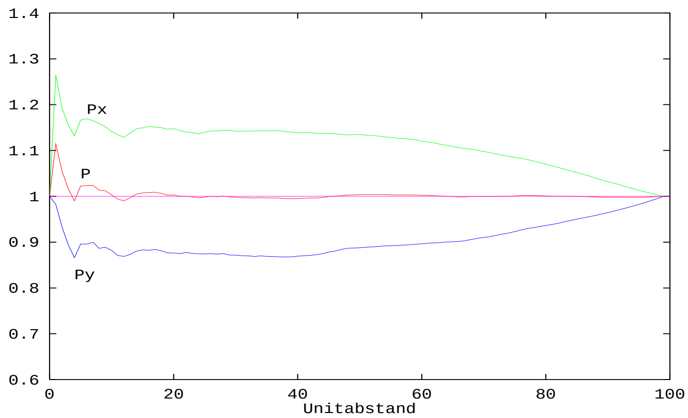


Fig. 3.11: Produktkurven zum verknoteten 10×10 Netz (Fig. 2.4)

3.4 Untersuchung der Kurven

In [5] werden Netze mit topologischem Defekt durch falsche Steuerung von σ nicht betrachtet, dort geht es ausschließlich darum, die Dimension von X zu erkennen. Dazu wird nur die Kurve $P(k)$ verwendet; die Autoren untersuchen nicht, ob sich aus dem Verlauf von $P_X(k)$ und $P_Y(k)$ andere Schlüsse ziehen lassen. Um einen Wert zu erhalten, an dem abgelesen werden kann, in welcher Relation die Dimension des Netzes und die des Raumes stehen, wurde $P(k)$ in Gleichung (17) von [5] wie folgt gemittelt ($N := |Y|$):

$$P = \frac{1}{N(N-1)} \sum_{r \in Y} \sum_{k=1}^{N-1} \log(P(r, n_k^Y(r)))$$

Natürlicherweise bietet es sich an, statt dieser Mittelung das Integral über die Kurve zu bilden:

$$\int P(k)\,dk := \frac{1}{2}(P(1) + P(N)) + \sum_{k=2}^{N-1} P(k).$$

Mit Hilfe des Integrals definiere ich nun einen Wert für die Dimension und einen für den topologischen Defekt:

$$\begin{aligned} p_{dim} &:= \frac{1}{N}\left(\int P(k)\,dk - N\right) \quad \text{und} \\ p_{top} &:= \frac{1}{N}\int P_X(k) - P_Y(k)\,dk. \end{aligned} \tag{11}$$

Das Integral in p_{dim} ist demnach die Fläche zwischen der Kurve P und 1, das Integral in p_{top} ist die Fläche zwichen den Kurven $P_X(k)$ und $P_Y(k)$. Da beide Integrale von der Gesamtanzahl der Units abhängen, wurden p_{dim} und p_{top} noch normiert.

Betrachten wir nun folgendes Kriterium für die Dimension des Netzes (vergleiche (10)):

$$
\begin{aligned}
p_{dim} > \epsilon_{dim} &\implies \text{„Dimension“ von } X < dim(Y): && \text{die Netzdimension ist zu groß,}\\
p_{dim} < -\epsilon_{dim} &\implies \text{„Dimension“ von } X > dim(Y): && \text{die Netzdimension ist zu klein,}\\
|p_{dim}| < \epsilon_{dim} &\implies \text{„Dimension“ von } X = dim(Y): && \text{die Netzdimension ist richtig.}
\end{aligned}
$$

$\epsilon_{dim} > 0$ ist ein kritischer Wert, der noch zu bestimmen ist.

Das Netz hat einen topologischen Defekt, obwohl die Dimension stimmt, wenn

$$|p_{dim}| < \epsilon_{dim} \text{ und } p_{top} > \epsilon_{top}.$$

Stimmt die Dimension nicht, so hat das Netz immer einen topologischen Defekt.

Die Werte von p_{dim} und p_{top} für die eben besprochenen Beispiele können Tab. 3.2 entnommen werden.

Fig.	3.7	3.8	3.9	3.10	3.11
p_{dim}	$-1.54\ 10^{-3}$	$-3.91\ 10^{-2}$	$2.33\ 10^{-2}$	$5.31\ 10^{-3}$	$2.59\ 10^{-3}$
p_{top}	$2.17\ 10^{-2}$	$2.69\ 10^{-1}$	$3.40\ 10^{-1}$	$2.82\ 10^{-2}$	$2.06\ 10^{-1}$

Tab. 3.2: p_{dim} und p_{top} für Fig. 3.7 – 3.11

Es ist zu erkennen, daß $|p_{dim}|$ für Fig. 3.7, 3.10 und 3.11 (also bei den drei zweidimensionalen Netzen) deutlich kleiner ist, als in den beiden anderen Fällen. p_{top} ist dagegen für Fig. 3.8, 3.9 und 3.11 besonders groß, also bei falscher Dimension und bei dem Knoten. Für ϵ_{dim} und ϵ_{top} haben sich folgende Werte als sinnvoll erwiesen:

$$\epsilon_{dim} = 10^{-2} \qquad \epsilon_{top} = 10^{-1}$$

Diese beiden Werte liefern nicht nur bei den in dieser Arbeit aufgeführten Beispielen die richtigen Ergebnisse, sondern auch bei allen weiteren von mir durchgeführten Simulationen.

3.5 Weitere Beispiele zur Dimensionserkennung

Bei den bisher betrachteten Beispielen war der Raum X von einfacher Struktur. Für P wähle ich für die nächsten beiden Beispiele nicht die Gleichverteilung auf $X = [0,1]^2$, sondern auf dem L-förmigen Raum $X_L := X \backslash \{(x,y) \in [0,1]^2 |\ x \geq 0.5 \ \vee \ y \geq 0.5\}$. Trainiert werden ein quadratisches Netz mit 100 Units und ein L-förmiges Netz mit 84 Units. Dieses geht durch Entfernen einiger Units aus dem quadratischen Netz hervor.

In Fig. 3.12 liegen einige Units nicht in X_L. Das liegt daran, daß diese von Werten wie $(0.5, 0.8)$ bzw. $(0.8, 0.5)$ hin- und hergezogen werden, da die Gewinnerunit für solche

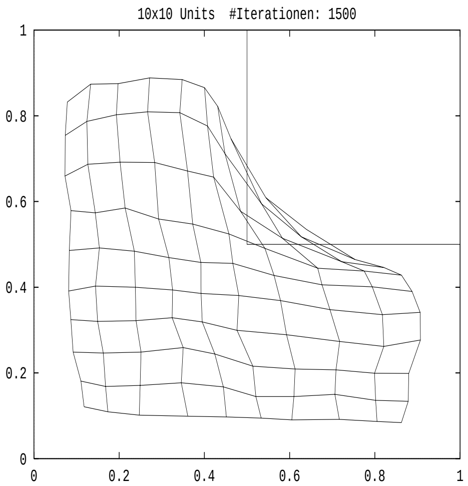


Fig. 3.12: ein quadratisches Netz

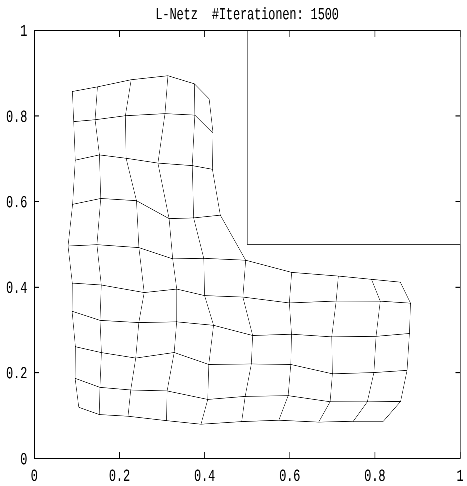


Fig. 3.13: ein L-förmiges Netz

Werte in ihrer Nähe liegt.

Offensichtlich ist das L-förmige Netz besser geeignet, um X_L ohne topologischen Defekt zu überdecken. Dieses läßt sich allerdings an den Kurven zu den beiden Netzen nur bedingt ablesen. Die in (11) definierten Werte sind für Fig. 3.14:

$$\begin{aligned} p_{dim} &= 3.09\ 10^{-3} \\ p_{top} &= 8.92\ 10^{-2} \end{aligned}$$

und für Fig. 3.15:

$$\begin{aligned} p_{dim} &= 2.52\ 10^{-3} \\ p_{top} &= 3.32\ 10^{-2}. \end{aligned}$$

Der Wert von p_{top} ist im ersten Fall nur etwas kleiner als der von ϵ_{top}. Der aufgetretende Defekt erscheint noch akzeptabel.

In [8] wird, mit einem anderen Ansatz, zusätzlich zu [5] auch die Form von X mit einbezogen.

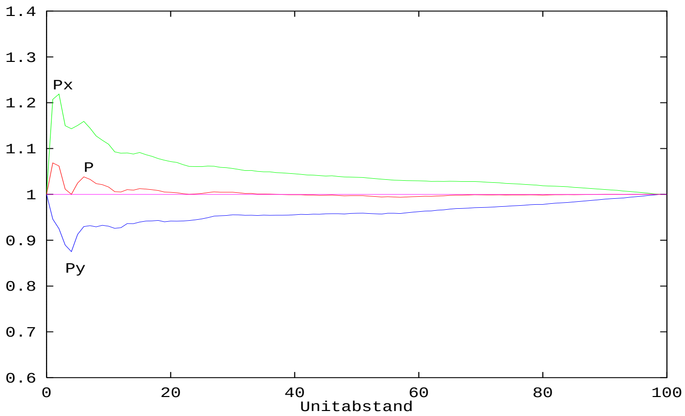


Fig. 3.14: Kurven zum quadratischen Netz (Fig. 3.12)

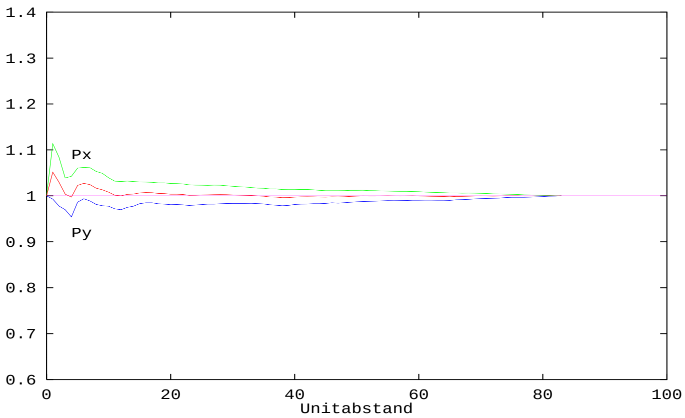


Fig. 3.15: Kurven zum L-förmigen Netz (Fig. 3.13)

Für die nächsten drei Beispiele sei $X \subset [0,1]^3$ die Oberfläche einer Kugel mit Mittelpunkt in $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ und Radius $\frac{1}{3}$. Ich möchte noch einmal darauf hinweisen, daß die w_{ij} bei der Initialisierung aus $[0,1]$ gewählt und nicht um den Mittelpunkt konzentriert werden (siehe 2.3 und 2.5).

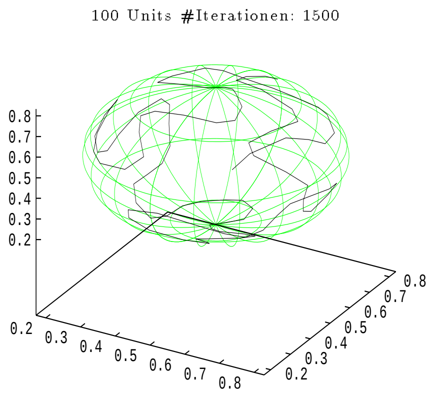


Fig. 3.16: ein eindimensionales Netz

In Fig. 3.16 ist ein eindimensionales Netz zu sehen, das sich auf der Sphäre gefaltet hat. Die Überschneidungen des Netzes sind in $\mathbb{R}^3$ nicht vorhanden, sie entstehen durch die Projektion ins Zweidimensionale.

Fig. 3.17 zeigt ein zweidimensionales Netz, das sich der Krümmung der Sphäre angepaßt hat.

In Fig. 3.18 ist schließlich ein dreidimensionales Netz zu sehen. Ebenso wie in Fig. 3.1 wurde das Netz wegen der Übersichtlichkeit nur zweidimensional verbunden. Während sich vier „Schichten" des Netzes der Krümmung der Sphäre angepaßt haben, verbleibt die fünfte relativ eben in der „Mitte". Units in dieser „Schicht" werden von den Daten der Sphäre in beide Richtungen „gezogen".

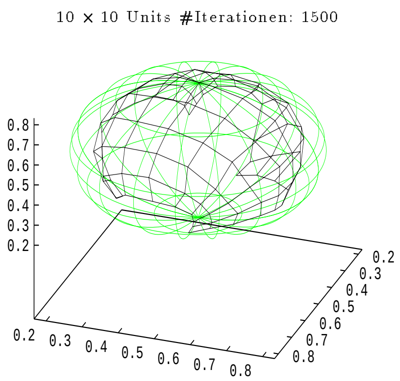


Fig. 3.17: ein zweidimensionales Netz

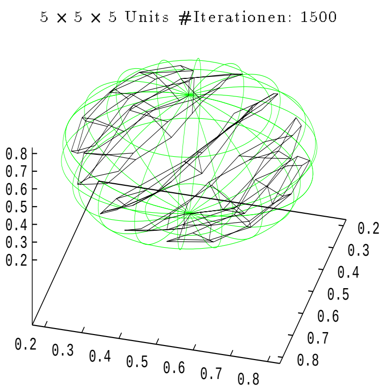


Fig. 3.18: ein dreidimensionales Netz

In Tab. 3.3 ist die Steuerung der Parameter ϵ und σ für das dreidimensionale Netz in Fig. 3.1 und 3.18 aufgeführt (siehe auch 2.7). Die maximale Entfernung D zwischen zwei Units beträgt für dieses Netz 15.

#Iterationen	25	50	100	300	500	> 500
$\sigma(t)$	6	5	4	3	2	1
$\epsilon(t)$	$\frac{1}{4}$		$\frac{1}{10}$			

Tab. 3.3: Parametersteuerung für ein $5 \times 5 \times 5$ Netz (Fig. 3.1 und 3.18)

In Fig. 3.19 ist die zu kleine Netzdimension durch den Verlauf von $P(k)$ und den Wert von p_{dim} deutlich zu erkennen:

$$p_{dim} = -5.40\ 10^{-2}$$
$$p_{top} = 2.84\ 10^{-1}.$$

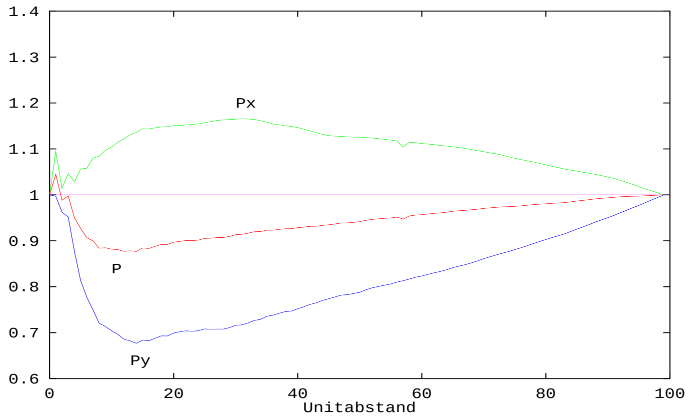


Fig. 3.19: Kurven zum eindimensionalen Netz (Fig. 3.16)

Die Kurven in Fig. 3.20 liegen am dichtesten an 1. Dementsprechend sind die Werte für die Dimension und Topologie auch am kleinsten:

$$p_{dim} = -1.90\ 10^{-3}$$
$$p_{top} = 6.56\ 10^{-2}.$$

Der relativ hohe Wert von p_{top} läßt sich durch die ziemlich ungleichmäßige Verteilung des Netzes erklären.

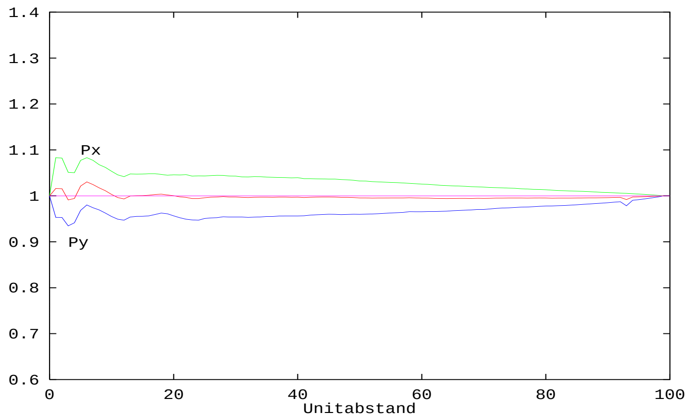


Fig. 3.20: Kurven zum zweidimensionalen Netz (Fig. 3.17)

In Fig. 3.21 signalisiert der Verlauf von $P_X(k)$ und $P_Y(k)$ einen topologischen Defekt. Insbesondere für kleinen Unitabstand ist eine starke Abweichung von 1 zu sehen. Diese kommt dadurch, daß die Units je zweier „Schichten“ einen deutlich kleineren Werteabstand haben, bei gleichem Unitabstand zur mittleren „Schicht“.

Diese große Abweichung fällt bei der Mittelung zu $P(k)$ weg. Die Werte sind:

$$p_{dim} = 7.02\ 10^{-3}$$
$$p_{top} = 1.15\ 10^{-1}.$$

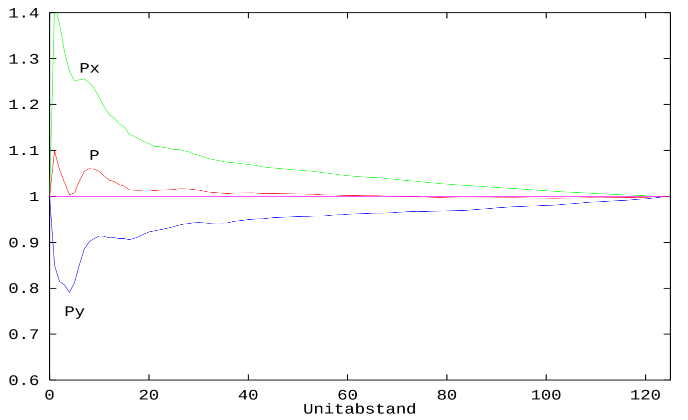


Fig. 3.21: Kurven zum dreidimensionalen Netz (Fig. 3.18)

Für die Spähre hat sich also ein zweidimensionales Netz deutlich als die beste Lösung unter den getesteten herausgestellt.

4. Nicht eindeutige Reihenfolge der Unitnachbarn

Wie schon in 3.1.1 bemerkt, ist die Definition der nächsten Unitnachbarn nicht eindeutig, da die Ortsvektoren der Units aus $\mathbb{N}_0^n$ gewählt werden. Abhängig von der Dimension n des Netzes und der Lage der betrachteten Unit gibt es m Unitnachbarn mit kleinstem Abstand, wobei $n \leq m \leq 2n$ gilt. Werden nicht die Nachbarn mit dem kleinsten Unitabstand betrachtet, dann kann die Anzahl der Units mit gleichem Abstand, falls diese Units am Rand des Netzes liegen, auch kleiner als n werden. Bei den Wertenachbarn ist die Reihenfolge praktisch immer eindeutig bestimmt, da die w_r kontinuierliche Werte annehmen.

Seien $n_k^Y(r)$ und $n_{k'}^Y(r)$ zwei Units mit gleichem Abstand zu Unit r, das heißt

$$\|r - n_k^Y\|_1 = \|r - n_{k'}^Y\|_1.$$

Dann ist es nicht festgelegt, welche der beiden Units in den in (7) definierten Quotienten $Q_X(r,k)$ und $Q_Y(r,k)$ verwendet wird. Durch diese nicht eindeutig bestimmte Reihenfolge gilt bei einer möglichen Vertauschung von $n_k^Y(r)$ und $n_{k'}^Y(r)$:

$$\begin{aligned} Q_Y(r,k) &= \frac{\|r - n_k^Y(r)\|_1}{\|r - n_k^X(r)\|_1} = \frac{\|r - n_{k'}^Y(r)\|_1}{\|r - n_k^X(r)\|_1} \quad \text{und} \\ Q_X(r,k) &= \frac{\|w_r - w_{n_k^Y(r)}\|_2}{\|w_r - w_{n_k^Y(r)}\|_2} \neq \frac{\|w_r - w_{n_{k'}^Y(r)}\|_2}{\|w_r - w_{n_k^Y(r)}\|_2}, \end{aligned}$$

da die beiden letzten Zähler im allgemeinen nicht übereinstimmen. Eine Vertauschung hat also nur Auswirkungen auf $Q_X(r,k)$ und nicht auf $Q_Y(r,k)$.

Nach Definition der Produkte (8) wirkt sich diese Veränderung von $Q_X(r,k)$ auf $P_X(r,k)$ und damit nach (9) auch auf $P(r,k)$ aus. Mit Hilfe der durch Mittelung über alle Units erhaltenen Kurve $P(k)$ wird entschieden, ob die Dimension des Netzes und die des Raumes übereinstimmen. Durch die Veränderung von $P(k)$ wird der Abstand zu 1 größer oder kleiner, je nach der vorherigen Lage von $P(k)$. Wie Beispiele zeigen, können nicht eindeutig bestimmte Unitnachbarn n^Y so angeordnet werden, daß die zugeordneten Kurven $P(k)$ sehr unterschiedlich verlaufen. Es ist dann nicht mehr möglich, eine gesicherte Aussage über die Relation der Dimensionen von Netz und Raum zu treffen ([7], siehe auch 4.4).

Die Autoren von [5] erwähnen in ihrer Veröffentlichung weder dieses Problem der nicht eindeutigen Reihenfolge noch die von ihnen benutzte „Lösung“. Diese besteht darin, Unitnachbarn mit gleichem Abstand jeweils *zufällig* anzuordnen ([7]).

4.1 Invarianten bei einer Änderung der Reihenfolge

Als eine *Sequenz* möchte ich im weiteren eine Folge aller Units bezeichnen, die denselben Unitabstand zu einer festen Unit r haben. Jedes Element der Folge aller Unitnachbarn $U(r)$ läßt sich mit Hilfe seines Abstands zu r einer Sequenz zuordnen. Jede Sequenz ist eine Teilfolge von $U(r)$. Die Reihenfolge der Units in einer Sequenz ist nicht eindeutig bestimmt, das heißt jede Permutation der Units ist zulässig.

Sei $S_k(r) = (n_k^Y(r), n_{k+1}^Y(r), \ldots, n_{k+m-1}^Y(r))$ eine Sequenz der Länge m. Mit einer Permutation der Elemente dieser Sequenz ändern sich die Werte von $Q_X(r,l)$ für $l = k, \ldots, k+m-1$ und somit auch die von $P_X(r,l)$ für $l = k, \ldots, k+m-2$.

Lemma 2: *Der Wert von $P_X(r, k+m-1)$ ist invariant unter jeder Permutation der Elemente von $S_k(r)$.*
Beweis: Sei $p := \prod_{l=1}^{k-1} Q_X(r,k)$ das Produkt der ersten $k-1$ Quotienten. Dann gilt:

$$P_X(r, k+m-1) = \left(p \prod_{l=k}^{k+m-1} Q_X(r,l)\right)^{\frac{1}{k+m-1}}.$$

Eine Permutation der Elemente der Sequenz bewirkt lediglich eine entsprechend andere Reihenfolge der Quotienten in dem Produkt. □

4.2 Die mögliche Abweichung für eine Sequenz

Sei r die im folgenden betrachtete Unit und $U(r)$ die Folge aller Unitnachbarn. Die durch eine Permutation in einer Sequenz $S_k(r)$ hervorgerufene Veränderung der zugehörigen Kurven $P_X(r,l)$ ist wegen der geometrischen Mittelung abhängig von k, das heißt von dem Index des ersten Elements von $S_k(r)$ in $U(r)$.

Seien zum Beispiel die Länge von $S_k(r)$ gleich vier und die zugehörigen Werte von $Q_X(r,k) = (1,2,3,4)$. Diese Schreibweise soll $Q_X(r,k) = 1$, $Q_X(r,k+1) = 2$, $Q_X(r,k+2) = 3$ und $Q_X(r,k+3) = 4$ bedeuten. $Q_X(r,k)$ bezeichnet also sowohl den k-ten Quotienten, als auch die Folge der $S_k(r)$ zugeordneten Quotienten, je nachdem ob $Q(r,k)$ eine Zahl oder eine Folge zugeordnet wird. Jede Permutation in einer Sequenz $S_k(r)$ induziert eine Permutation in der zugehörigen Folge $Q_X(r,k)$. Ich betrachte die beiden folgenden Permutationen in $Q_X(r,k)$ für zwei verschiedene k:

$Q_X(r,k) = (1,2,3,4)$	$Q_X(r,k) = (4,3,2,1)$
1. $k = 1$	
$\Longrightarrow P_X(r,1) = (1, \sqrt{2}, \sqrt[3]{6}, \sqrt[4]{24})$	$\Longrightarrow P_X(r,1) = (4, \sqrt{12}, \sqrt[3]{24}, \sqrt[4]{24})$
$\Longrightarrow \int_1^4 P_X(r,l)dl \approx 4.838$	$\Longrightarrow \int_1^4 P_X(r,l)dl \approx 9.455$
2. $k = 5$ (Ann.: $P_X(r,4) = 1$)	
$\Longrightarrow P_X(r,5) = (1, \sqrt[6]{2}, \sqrt[7]{6}, \sqrt[8]{24})$	$\Longrightarrow P_X(r,5) = (\sqrt[5]{4}, \sqrt[6]{12}, \sqrt[7]{24}, \sqrt[8]{24})$
$\Longrightarrow \int_5^8 P_X(r,l)dl \approx 3.658$	$\Longrightarrow \int_5^8 P_X(r,l)dl \approx 4.491$

Ein Quotient $Q_X(r,k)$ kann nach Definition beliebig groß werden. Bei den im letzten Kapitel gerechneten Beispielen lagen die Werte der Quotienten fast ausschließlich im Intervall $[0,5]$. Die Annahme $P_X(r,4) = 1$ wurde getroffen, da nur die Abweichung für eine Sequenz untersucht wird. Ist $P_X(r,4) > 1$, dann erhöhen sich die Werte für beide Integrale. Dadurch wird die Differenz der Integrale für die beiden Permutationen größer.

An den Beispielen sind zwei Dinge zu erkennen:

1. Der Wert des Integrals über die Kurve $P_X(r,k)$ ist erheblich abhängig von der gewählten Permutation in der Sequenz.
2. Die Abweichung der zwei Integrale ist wesentlich abhängig von dem Index des ersten Elements der Sequenz in $U(r)$.

Seien nun $Q^1_X(r,1) = (q_1, q_2, q_3, q_4)$ mit $q_1 \geq q_2 \geq q_3 \geq q_4 > 0$ die Werte für eine Sequenz in $U(r)$ mit Länge vier. Ich betrachte diese und die durch eine Permutation in der Sequenz erzeugte Reihenfolge $Q^2_X(r,1) = (q_4, q_2, q_3, q_1)$. Dann gilt:

$$P^1_X(r,1) = (q_1, \sqrt{q_1 q_2}, \sqrt[3]{q_1 q_2 q_3}, \sqrt[4]{q_1 q_2 q_3 q_4}) \text{ und}$$
$$P^2_X(r,1) = (q_4, \sqrt{q_4 q_2}, \sqrt[3]{q_4 q_2 q_3}, \sqrt[4]{q_4 q_2 q_3 q_1}).$$

$$\begin{aligned}
\Longrightarrow \quad \int_1^4 P^1_X(r,l)\,dl &= \frac{1}{2}(q_1 + \sqrt[4]{q_1 q_2 q_3 q_4}) + \sqrt{q_1 q_2} + \sqrt[3]{q_1 q_2 q_3} \\
&\leq \frac{1}{2}(q_1 + q_1) + q_1 + q_1 = 3q_1 \quad \text{und} \\
\int_1^4 P^2_X(r,l)\,dl &= \frac{1}{2}(q_4 + \sqrt[4]{q_4 q_2 q_3 q_1}) + \sqrt{q_4 q_2} + \sqrt[3]{q_4 q_2 q_3}
\end{aligned}$$

$$\begin{aligned}
\Longrightarrow \quad \int_1^4 P^1_X(r,l) - P^2_X(r,l)\,dl &= \frac{1}{2}(q_1 - q_4) + \sqrt{q_2}(\sqrt{q_1} - \sqrt{q_4}) + \sqrt[3]{q_2 q_3}(\sqrt[3]{q_1} - \sqrt[3]{q_4}) \\
&\geq \frac{1}{2}(q_1 - q_4)
\end{aligned}$$

$$\Longrightarrow \quad \frac{\int_1^4 P^1_X(r,l) - P^2_X(r,l)\,dl}{\int_1^4 P^1_X(r,l)\,dl} \geq \frac{\frac{1}{2}(q_1 - q_4)}{3q_1} = \frac{1}{6}\left(1 - \frac{q_4}{q_1}\right)$$

Schon durch eine Vertauschung von zwei Elementen entsteht für eine Unit eine relative Verkleinerung des Integrals um mindestens 13% für den oft auftretenden Fall $q_1 \geq 4q_4$.

4.3 Schranken für die mögliche Abweichung

Im folgenden wird untersucht, wie stark die Kurven $P_X(k)$ und $P(k)$ für verschiedene Permutationen in den Sequenzen variieren können. Sei $S_k(r) = (n^Y_k(r), \ldots, n^Y_{k+m-1}(r))$ wieder eine Sequenz mit Startindex k und Länge m. Für diese Sequenz betrachte ich die zwei Permutationen, welche die der Sequenz zugeordneten Quotienten $Q_X(r,k)$ einmal aufsteigend und einmal absteigend anordnen:

$$\begin{aligned}
Q^{min}_X(r,k) &= (q_k, \ldots, q_{k+m-1}) \\
Q^{max}_X(r,k) &= (q_{k+m-1}, \ldots, q_k)
\end{aligned} \quad \text{mit } q_k \leq \ldots \leq q_{k+m-1}.$$

Analog zur Definition (8) erhält man die Funktionen $P_X^{min}(r,k)$ bzw. $P_X^{max}(r,k)$, indem für alle Sequenzen die beiden Permutationen gewählt werden, die die zugeordneten Quotienten aufsteigend bzw. absteigend anordnen. Beide Funktionen werden, wie im letzten Kapitel für die Bilder, über alle Units r arithmetisch gemittelt. Dann gilt

Satz 1: *Die Funktionen $P_X^{min}(k)$ bzw. $P_X^{max}(k)$ sind eine untere bzw. obere Schranke für jede, durch beliebige Permutationen in den Sequenzen zu erhaltende Funktion $P_X(k)$:*

$$\forall\, k: \quad P_X^{min}(k) \le P_X(k) \le P_X^{max}(k).$$

Beweis: Zunächst betrachte ich die Funktionen vor der Mittelung über alle Units. Sei $Q_X(r,k)$ durch eine beliebige Permutation der Sequenz $S_k(r)$ bestimmt, das heißt $Q_X(r,k) = (q_{i(0)}, \ldots, q_{i(m-1)})$ mit $k \le i(n) \le k+m-1 \;\; \forall\, n$ und $i(l) \ne i(l')$ für $l \ne l'$. Dann gilt wegen der Anordnung von $Q_X^{min}(r,k)$ und $Q_X^{max}(r,k)$:

$$q_k \le q_{i(0)} \le q_{k+m-1} \quad \text{und} \quad q_k q_{k+1} \le q_{i(0)} q_{i(1)} \le q_{k+m-1} q_{k+m-2}$$

oder allgemein für jedes Teilprodukt

$$\prod_{j=0}^{l} q_{k+j} \le \prod_{j=0}^{l} q_{i(j)} \le \prod_{j=0}^{l} q_{k+m-j-1} \quad \text{für } 0 \le l \le m-1.$$

Wegen der Monotonie der Wurzel gilt weiter

$$\left(\prod_{j=0}^{l} q_{k+j}\right)^{\frac{1}{k+l}} \le \left(\prod_{j=0}^{l} q_{i(j)}\right)^{\frac{1}{k+l}} \le \left(\prod_{j=0}^{l} q_{k+m-j-1}\right)^{\frac{1}{k+l}} \quad \text{für } 0 \le l \le m-1. \qquad (12)$$

In Lemma 2 wurde bereits gezeigt, daß der Wert von $P_X(r,j)$ invariant unter jeder Permutation ist, falls bei j eine Sequenz endet. Das heißt insbesondere, da für $k > 1$ an der Stelle $k-1$ das Ende der vorhergehenden Sequenz ist, daß der Wert von $P_X(r,k-1)$ invariant unter jeder Anordnung in den Sequenzen mit Startindex kleiner als k ist. Sei $p := P_X(r,k-1)^{k-1}$ das Produkt der ersten $k-1$ Quotienten ($p := 1$ für $k = 1$). Dann ist (12) äquivalent mit

$$\begin{array}{lllll} & \left(p\prod_{j=0}^{l} q_{k+j}\right)^{\frac{1}{k+l}} \le & \left(p\prod_{j=0}^{l} q_{i(j)}\right)^{\frac{1}{k+l}} & \le \left(p\prod_{j=0}^{l} q_{k+m-j-1}\right)^{\frac{1}{k+l}} & 0 \le l \le m-1 \\ \iff & P_X^{min}(r,l) \le & P_X(r,l) & \le P_X^{max}(r,l) & k \le l \le k+m-1 \end{array}$$

Der Startindex k der Sequenz war beliebig gewählt. Deswegen läßt sich dieselbe Schlußweise auf alle anderen Sequenzen in $U(r)$ anwenden.

$$\Longrightarrow \quad \forall\, k: \quad P_X^{min}(r,k) \le P_X(r,k) \le P_X^{max}(r,k)$$

Diese Ungleichungen gelten für alle Units $r \in Y$.

$$\Longrightarrow \quad \forall\, k: \quad \frac{1}{|Y|}\sum_{r\in Y} P_X^{min}(r,k) \leq \frac{1}{|Y|}\sum_{r\in Y} P_X(r,k) \leq \frac{1}{|Y|}\sum_{r\in Y} P_X^{max}(r,k)$$

$$\stackrel{\text{def}}{\Longleftrightarrow} \quad \forall\, k: \quad P_X^{min}(k) \leq P_X(k) \leq P_X^{max}(k) \qquad \square$$

Folgerung 1: *Seien $P^{min}(k)$ und $P^{max}(k)$ analog zu $P(k)$ in (9) definiert, das heißt unter Verwendung von $P_X^{min}(k)$ bzw. $P_X^{max}(k)$ statt $P_X(k)$. Dann gilt:*

$$\forall\, k: \quad P^{min}(k) \leq P(k) \leq P^{max}(k).$$

Beweis: Da $P_X^{min}(k)$ und $P_X^{max}(k)$ nach Satz 1 eine untere bzw. obere Schranke für alle durch Permutationen in Sequenzen zu erhaltenden Funktionen $P_X(k)$ sind, folgt die Aussage unmittelbar aus Definition (9).

Bemerkung 1: Die in Lemma 2 gefundene Invarianz von $P_X(r,k_e)$ gegenüber Permutationen bedeutet, daß für jeden Endindex k_e einer Sequenz insbesondere $P_X^{min}(r,k_e) = P_X(r,k_e) = P_X^{max}(r,k_e)$ gilt, unabhängig davon, mit welcher Permutation $P_X(r,k_e)$ gewonnen wird.

Bei der Mittelung über alle Units geht diese Eigenschaft verloren, da die Sequenzen für jede Unit im allgemeinen verschiedene Längen und damit andere Start- und Endindizes haben. Deswegen gibt es im allgemeinen keinen Index, bei dem für alle Units eine Sequenz endet.

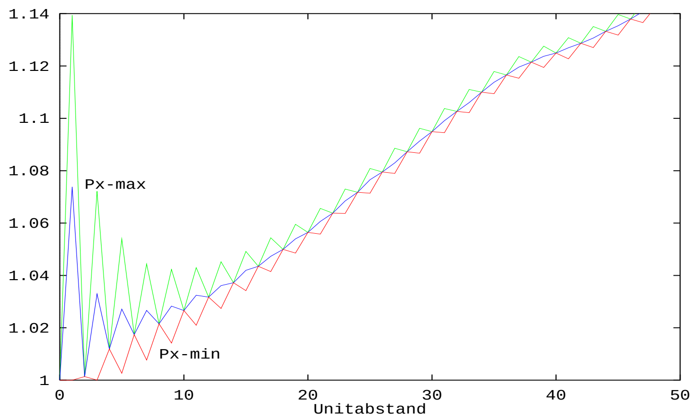


Fig. 4.1: P_X Kurven für das 256 Netz (vergleiche Fig. 3.8)

Fig. 4.1 zeigt für das eindimensionale Netz im zweidimensionalen Raum (Beispiel 2) die Kurven $P_X^{min}(k)$ und $P_X^{max}(k)$ für $k \leq 50$ und zwischen beiden die Kurve $P_X(k)$ aus Fig. 3.8 mit zufälliger Anordnung in den Sequenzen. Die in Lemma 2 gefundene Invarianz ist deutlich zu sehen, da die drei Kurven für alle geraden k zusammenfallen. Bei dem eindimensionalen Netz treten nur Sequenzen der Länge eins und zwei auf. Deswegen bleibt die Invarianz bei der Mittelung über alle Units erhalten.

In Fig. 4.1 ist weiterhin zu erkennen, daß die Kurve $P_X(k)$ ungefähr das arithmetische Mittel aus $P_X^{min}(k)$ und $P_X^{max}(k)$ ist. Diese Tatsache hängt unmittelbar damit zusammen, daß zur Bestimmung von $P_X(k)$ die Reihenfolge der Elemente jeder Sequenz zufällig bestimmt wurde (siehe 3.1.1). Deswegen ist das arithmetische Mittel aus $P_X^{min}(k)$ und $P_X^{max}(k)$ der Erwartungswert für $P_X(k)$.

4.4 Untersuchung der Beispiele aus 3.3 und 3.5

Liegt $P_X(k)$ schon sehr dicht bei 1, wie bei den beiden quadratischen Netzen (Fig. 3.7 und Fig. 3.10), so ist auch der Abstand zwischen $P_X^{min}(k)$ und $P_X^{max}(k)$ klein. Beide Funktionen haben, da der Erwartungswert für $P_X(k)$ das arithmetische Mittel aus $P_X^{min}(k)$ und $P_X^{max}(k)$ ist, punktweise ungefähr denselben Abstand von $P_X(k)$. Aus Lemma 1 folgt $\forall\ k\ :\ P_X^{min}(k), P_X^{max}(k) \geq 1$. Liegt $P_X(k)$ nun schon nahe bei 1, so kann der Abstand zu $P_X^{min}(k)$ und somit auch der zwischen $P_X^{min}(k)$ und $P_X^{max}(k)$ nicht groß sein. Die Kurven von $P_X^{min}(k)$ und $P_X^{max}(k)$ werden für diese beiden Beispiele nicht besprochen, da sie fast mit $P_X(k)$ zusammenfallen.

Auf den interessanten Teil der Kurven $P_X^{min}(k)$ und $P_X^{max}(k)$ für das eindimensionale Netz (Fig. 4.1) wurde eben schon eingegangen. Der Abstand der beiden Kurven und damit auch der von $P^{min}(k)$ und $P^{max}(k)$ ist für $k > 10$ sehr klein.

Die Differenz ist dagegen bei den beiden verbliebenen Beispielen aus 3.3 wesentlich größer. Die in den folgenden Bildern zu sehenden Kurven $P_X(k)$ und $P(k)$ entsprechen jeweils denen in den zugehörigen Abbildungen aus Kapitel 3. Dies ist wegen der anderen Skalierung nicht direkt zu sehen.

Zunächst möchte ich die Kurven zum verknoteten Netz (Fig. 3.11) untersuchen. Als erstes ist zu erkennen, daß die Invarianz nach Lemma 2 bei der Mittelung über alle Units verloren geht (siehe Bem. 1). Die Kurven haben nur noch Start- und Endpunkt gemein (Fig. 4.2).

Für kleinen Unitabstand variieren die Kurven besonders stark. Dann ist der mögliche Abstand von $P_X^{min}(k)$ und $P_X^{max}(k)$, bedingt durch die geometrische Mittelung der Produkte größer (siehe auch 4.2). Der Abstand der beiden Kurven voneinander ist insgesamt so groß, daß sich für die zugehörigen Kurven P ein unterschiedlicher Verlauf zeigt (Fig. 4.3).

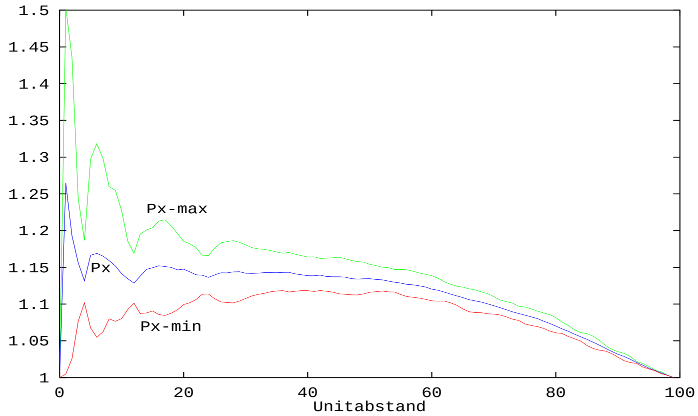


Fig. 4.2: P_X Kurven zu Fig. 3.11

	min	Fig. 3.11	max
p_{dim}	$-1.05\ 10^{-2}$	$2.59\ 10^{-3}$	$1.68\ 10^{-2}$
p_{top}	$1.77\ 10^{-1}$	$2.06\ 10^{-1}$	$2.38\ 10^{-1}$

Tab. 4.1: p_{dim} und p_{top} zu Fig. 4.2 und 4.3

$P^{max}(k)$ verläuft ähnlich wie $P(k)$, nur mit größeren Maxima. Dagegen liegt $P^{min}(k)$ durchgehend unterhalb von 1, was eine zu kleine Dimension des Netzes signalisiert.

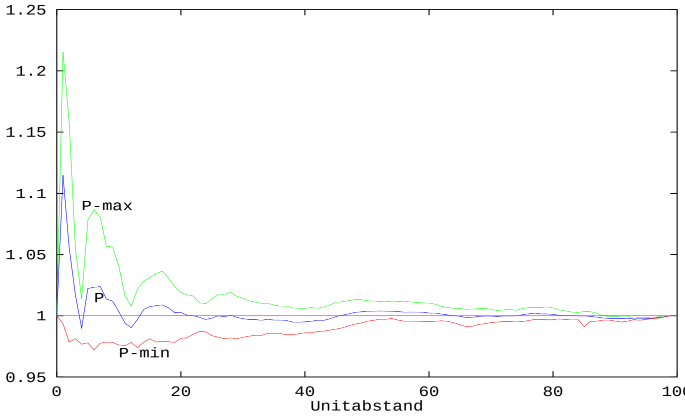


Fig. 4.3: P Kurven zu Fig. 3.11

Das ist auch an den in (3.4) definierten Werten p_{dim} und p_{top} zu sehen, die in Tab. 4.1 aufgeführt sind. Für das dreidimensionale Netz aus dem letzten Kapitel gilt $p_{dim} = 2.33\ 10^{-2}$ (siehe Tab. 3.2, S. 26), p_{dim}^{max} aus diesem Beispiel ist mehr als halb so groß. Es ist fraglich, ob unter diesen Umständen noch eine verläßliche Aussage über die Dimension getroffen werden kann. p_{top} variiert relativ wenig, was vor allem daran liegt, daß die Kurve $P_Y(k)$, die schon in Fig. 3.11 recht weit von 1 entfernt ist, durch die Permutation in den Sequenzen nicht verändert wird.

Bei dem nächsten Beispiel, dem $5\times5\times5$ Netz, sind die Auswirkungen durch die Permutation in den Sequenzen noch deutlicher. Die in Fig. 4.4 zu sehenden Kurven haben keinen besonders großen Abstand voneinander. Allerdings verläuft die Kurve $P(k)$ in Fig. 3.9 relativ nahe bei 1, das heißt schon kleine Schwankungen von $P_X(k)$ können dafür sorgen, daß $P(k)$ unterhalb von 1 verläuft.

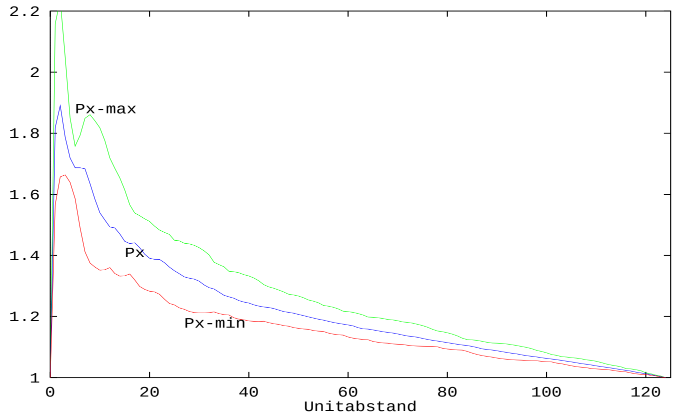


Fig. 4.4: P_X Kurven zu Fig. 3.9

Genau dieses ist in Fig. 4.5 zu beobachten. $P^{min}(k)$ verläuft sowohl unter- als auch oberhalb von 1, weswegen p_{dim}^{min} sehr klein ist (siehe Tab. 4.2). Dabei ist zu beachten, daß p_{dim} für das 5×5 Netz sogar noch größer ist (siehe Tab. 3.2, Fig. 3.10). Es

	min	Fig. 3.9	max
p_{dim}	$1.74\ 10^{-3}$	$2.33\ 10^{-2}$	$4.87\ 10^{-2}$
p_{top}	$2.84\ 10^{-1}$	$3.40\ 10^{-1}$	$4.08\ 10^{-1}$

Tab. 4.2: p_{dim} und p_{top} zu Fig. 4.4 und 4.5

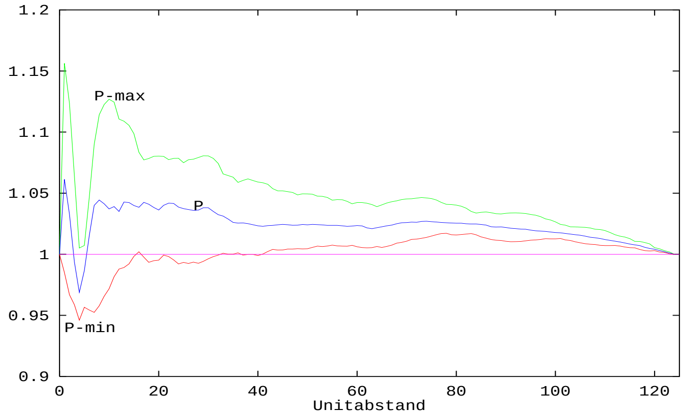


Fig. 4.5: P Kurven zu Fig. 3.9

läßt sich also kein Dimensionsunterschied zwischen dem zwei- und dreidimensionalen Netz bei demselben Datenraum feststellen, falls $P^{min}(k)$ für das 5×5×5 Netz gewählt wird.

Die Wahl, die Reihenfolge in den Sequenzen zufällig zu bestimmen, ist in [5], ebenso wie die Dimensionentscheidung, mathematisch nicht gerechtfertigt. Genauso gut kann jeweils die Permutation gewählt werden, die die maximale bzw. minimale Kurve erzeugt. Die Dimensionsentscheidung wird problematisch, falls die verschiedenen Kurven nicht alle oberhalb, unterhalb oder nahe bei 1 verlaufen. In diesem Fall ergibt die Interpretation der Kurven für ein und dasselbe Netz eine unterschiedliche Relation zwischen der Dimension des Netzes und der des Raumes.

Zwei weitere Beispiele dafür sind das zwei- und das dreidimensionale Netz mit der Sphäre als Datenraum. Bei dem eindimensionalen Netz (Fig. 3.16 und 3.19) kommt es kaum zu Abweichungen von $P_X(k)$ und $P(k)$, ähnlich wie in Fig. 4.1.

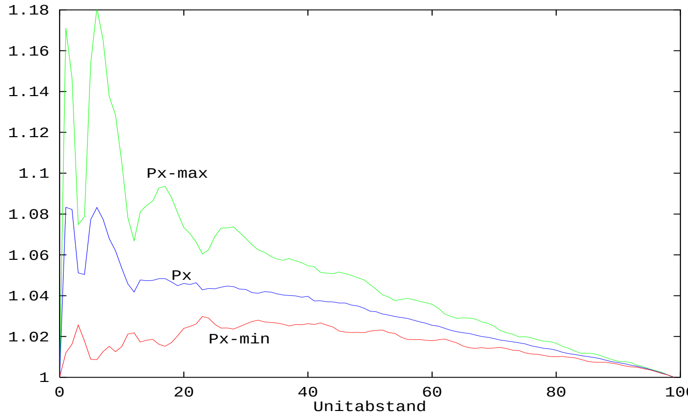


Fig. 4.6: P_X Kurven zu Fig. 3.20

Zunächst zum zweidimensionalen Netz (Fig. 3.17):
Die möglichen Abweichungen für die Kurve $P_X(k)$ sind in Fig. 4.6 zu sehen. $P_X^{max}(k)$ und $P_X^{min}(k)$ liegen insbesondere für kleine k weit auseinander. Zusammen mit dem kleinen Wert von p_{top} bei zufälliger Anordnung in den Sequenzen läßt sich die im Vergleich zu dem gleichgroßen verknoteten Netz (siehe Tab. 4.1) große Differenz von p_{top}^{min} und p_{top}^{max} erklären.

	min	Fig. 3.20	max
p_{dim}	$-8.92\ 10^{-3}$	$-1.90\ 10^{-3}$	$6.68\ 10^{-3}$
p_{top}	$5.08\ 10^{-2}$	$6.56\ 10^{-2}$	$8.35\ 10^{-2}$

Tab. 4.3: p_{dim} und p_{top} zu Fig. 4.6 und 4.7

Die Kurven $P^{max}(k)$ und $P(k)$ verlaufen beide ober- und unterhalb von 1, dagegen liegt $P^{min}(k)$ durchgehend unterhalb von 1 (Fig. 4.7). Die Werte von $|p_{dim}|$ sind kleiner als der von mir gesetzte kritische Wert von 10^{-2}. Die Werte von $|p_{dim}^{min}|$ und $|p_{dim}^{max}|$ liegen allerdings nur knapp unterhalb der Schranke.

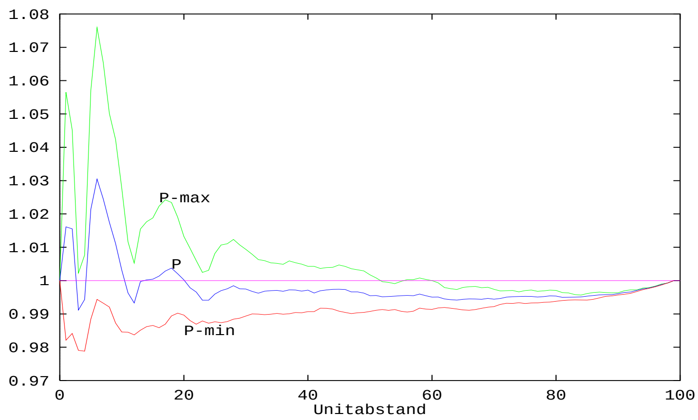


Fig. 4.7: P Kurven zu Fig. 3.20

Noch deutlicher wird die Problematik bei dem dreidimensionalen Netz. Obwohl $P_X^{min}(k)$ erheblich näher an 1 verläuft als $P_X(k)$, ist p_{top}^{min} nur etwas kleiner als p_{top} (siehe Tab. 4.4). Das liegt, wie schon erläutert daran, daß $P_Y(k)$ durch die Permutationen nicht verändert wird.

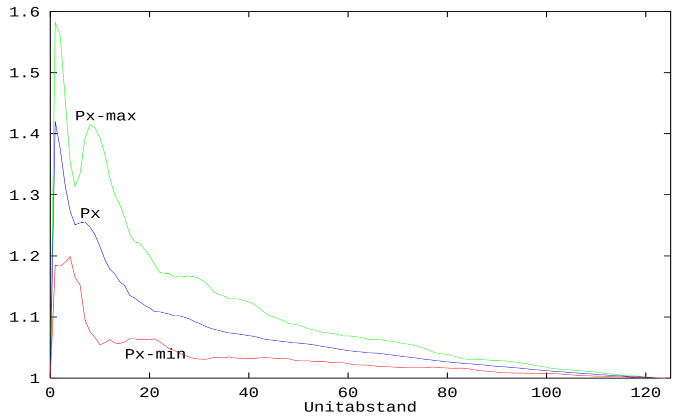


Fig. 4.8: P_X Kurven zu Fig. 3.21

Für die Kurven $P(k)$ zeigt sich ein ähnliches Bild wie bei dem zweidimensionalen Netz, nur daß $P^{max}(k)$ und $P(k)$ noch deutlicher oberhalb von 1 verlaufen. Die Werte von p_{dim} verbieten jegliche Aussage darüber, ob die Netzdimension korrekt ist. Für dasselbe Netz mit dem zweidimensionalen Raum ist p_{dim} mit $2.33\ 10^{-2}$ (siehe Tab. 3.2, Fig. 3.9) sogar noch kleiner, als p_{dim}^{max} in diesem Beispiel.

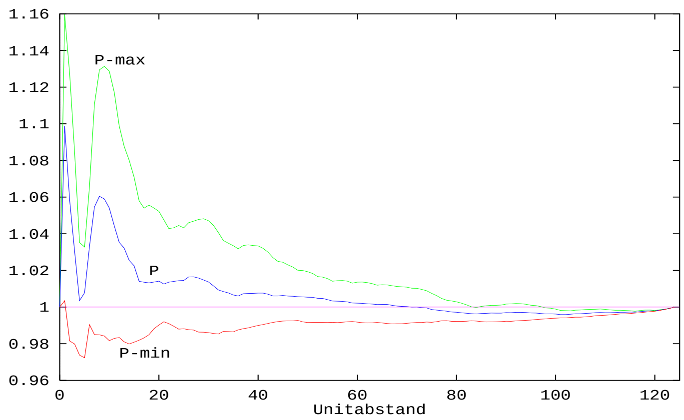


Fig. 4.9: P Kurven zu Fig. 3.21

	min	Fig. 3.21	max
p_{dim}	$-9.12\ 10^{-3}$	$7.02\ 10^{-3}$	$2.49\ 10^{-2}$
p_{top}	$7.92\ 10^{-2}$	$1.15\ 10^{-1}$	$1.55\ 10^{-1}$

Tab. 4.4: p_{dim} und p_{top} zu Fig. 4.8 und 4.9

In [5] wird die in Kapitel 3 beschriebene Methode auf Sprachdaten angewendet. Betrachtet man sich das Bild in [5] mit den Kurven $P(k)$ für die verschiedenen Netzdimensionen (Fig. 4.10), so ist zu sehen, daß der Abstand der Kurven des drei- bzw. vierdimensionalen Netzes relativ klein ist. Wird statt der zufälligen Anordnung in den Sequenzen die für $P^{min}(k)$ gewählt, dann werden alle Kurven nach unten verschoben. Die Kurve des vierdimensionalen Netzes verliefe dann, wenn eine ähnliche Abweichung wie in Fig. 4.9 unterstellt wird, unter- und oberhalb von 1. Dann wäre mit der Argumentation aus [5] das Ergebnis gewesen: Sprachdaten können in vierdimensionalen Netzen am besten gespeichert werden.

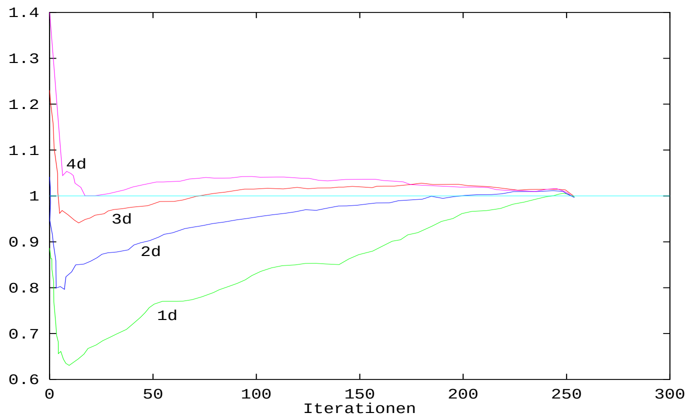


Fig. 4.10: P Kurven für Sprachdaten ([5], Fig 13)

4.5 Vorschlag für eine Lösung des Problems der nicht eindeutigen Reihenfolge

Da jede Anordnung der Elemente einer Sequenz gewählt werden kann, ist das Ergebnis mehr oder weniger willkürlich. Es bietet sich folgender Ausweg an: Statt ausschließlich p_{dim} bei der Entscheidung über die Dimension zu betrachten, werden p_{dim}^{min} und p_{dim}^{max} betrachtet. Das so modifizierte Kriterium lautet (vergleiche S. 26):

$$p_{dim}^{min} > \epsilon_{dim} \implies \text{„Dimension“ von } X < dim(Y): \text{ die Netzdimension ist zu groß,}$$

$$p_{dim}^{max} < -\epsilon_{dim} \implies \text{„Dimension“ von } X > dim(Y): \text{ die Netzdimension ist zu klein,}$$

$$\left.\begin{array}{l} |p_{dim}^{min}| < \epsilon_{dim} \\ |p_{dim}^{max}| < \epsilon_{dim} \end{array}\right. \implies \text{„Dimension“ von } X = dim(Y): \text{ die Netzdimension ist richtig.}$$

Tritt keiner der drei Fälle ein, so ist keine qualifizierte Aussage über die Relation der Dimension des Netzes und der des Raumes möglich. Anschaulich bedeutet dieses Kriterium:

1. Verläuft $P^{min}(k)$ hinreichend deutlich oberhalb von 1, dann ist die Netzdimension zu groß.
2. Verläuft $P^{max}(k)$ hinreichend deutlich unterhalb von 1, dann ist die Netzdimension zu klein.
3. Verlaufen beide Kurven hinreichend nahe bei 1, dann stimmt die Dimension des Netzes und die des Raumes überein.
4. In allen anderen Fällen ist keine gesicherte Aussage möglich, wie in den besprochenen Beispielen.

Der Begriff „hinreichend“ wird durch ϵ_{dim} quantifiziert.

Auch dieses Kriterium ist mathematisch nicht gesichert. Es werden dadurch lediglich die durch die nicht eindeutige Definition der nächsten Nachbarn erzeugten Probleme behoben. Das Kriterium ist nach wie vor nur durch viele Beispiele motiviert, bei denen es das richtige Ergebnis liefert.

5. Zusammenfassung

In dieser Arbeit wurde das Modell von T. Kohonen für selbstorganisierende künstliche Neuronale Netze ausführlich dargestellt. Der Unitabstand und die Nachbarschaftsfunktion dieses Modells werden anders als üblich gewählt, um eine gute interaktive Steuerung des Netzes zu ermöglichen. Bei der Besprechung zweier Standardbeispiele stellt sich heraus, daß es ausreicht, das Netz bei der Initialisierung zufällig im ganzen Raum zu verteilen und es nicht einschränkend um dessen Mittelpunkt zu konzentrieren, wie es in vielen Veröffentlichungen geschieht. Die von mir selbst entwickelte Fehlerfunktion für ein selbstorganisierendes Netz zeigt die Ausbreitung des Netzes im Raum und die Überdeckung des Raumes an, nicht jedoch einen topologischen Defekt des Netzes. Schließlich wird für alle besprochenen Beispiele die Steuerung der zwei zu wählenden Parameter angegeben. Diese wird in fast keiner einschlägigen Publikation erwähnt.

Durch Überprüfung der in dem Modell von Kohonen geforderten Nachbarschaftserhaltung läßt sich die „Dimension" der Eingabedaten erkennen. Die dafür von H.-U. Bauer und K.R. Pawelzik in [5] entwickelte Lösung wird ausführlich besprochen. Das von ihnen vorgeschlagene Kriterium wird allerdings durch ein plausibleres ersetzt. Es stellt sich zusätzlich zu [5] heraus, daß es mit dieser Lösung möglich ist, einen topologischen Defekt des Netzes zu erkennen. Die Grenzen der in [5] vorgeschlagenen Lösung werden durch neue Beispiele mit noch nicht betrachteten Datenräumen deutlich.

Der grundsätzliche Mangel des Ansatzes von [5], die nicht eindeutige Definition der Nachbarn einer Unit, wird im weiteren untersucht. In einem Satz werden die obere und untere Schranke, die beide erreicht werden können, für den dadurch hervorgerufenen Fehler angegeben. Der Fehler ist in einigen Beispielen aus [5] so groß, daß die „Dimension" der Eingabedaten nicht genau ermittelt werden kann. Schließlich wird als Lösung ein modifiziertes Kriterium vorgeschlagen. Durch dieses wird für viele Probleme die Nichtentscheidbarkeit der Dimensionsfrage mit dem Ansatz aus [5] erkannt. Insbesondere ist es fraglich, ob die Aufsehen erregenden Ergebnisse zu den Sprachdaten in [5] noch haltbar sind.

Literatur

[1] H. Ritter, T. Martinetz, K. Schulten: *Neuronale Netze*, Addison-Wesley, 2. Aufl. 1991

[2] T. Kohonen: *Self-organization and associative Memory*, Springer, 3rd ed. 1989

[3] J. Hertz, A. Krogh, R.G. Palmer: *Introduction to the theory of neural computing*, Addison-Wesley, 1st ed. 1991

[4] R. Beale, T. Jackson: *Neural Computing*, Adam Hilger, 1st ed. 1990

[5] H.-U. Bauer, K. R. Pawelzik: *Quantifying the Neighbourhood Preservation of Self-Organizing Feature Maps*, IEEE Transactions on Neural Networks, Vol. 3, No. 4, July 1992

[6] W. Liebert, K. Pawelzik, H.G. Schuster: *Optimal Embedding of Chaotic Attractors from Topological Considaration*, Europhysical Letters 14 (6), March 1991

[7] Gespräch mit H.-U. Bauer am 19.4.1995

[8] Th. Villmann, R. Der, M. Herrmann, Th. Martinetz: *Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement*, Preprint 11/94, Naturwissenschaftliches Zentrum der Universität Leipzig