



TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Vorhersage der Erdrotationsparameter mit Hilfe Neuronaler Netze

M. Ulrich

IAPG / FESG No. 9

Institut für Astronomische und Physikalische Geodäsie
Forschungseinrichtung Satellitengeodäsie

München 2000

Vorhersage der Erdrotationsparameter mit Hilfe Neuronaler Netze

Markus Ulrich

26. Mai 2000

Inhaltsverzeichnis

1	Einleitung	2
2	Grundlagen	4
2.1	Erdrotationsparameter	4
2.1.1	Zeitskalen und Definitionen	4
2.1.2	Geophysikalische Zusammenhänge	6
2.1.3	Anwendung der Erdrotationsparameter	8
2.2	Neuronale Netze	10
2.2.1	Arbeitsweise des Gehirns	10
2.2.2	Mathematische Darstellung	13
2.2.3	Vor- und Nachteile	20
3	Beschreibung von SNNS	22
3.1	Bestandteile des Software-Paketes <i>SNNS</i>	22
3.1.1	Graphische Oberfläche	22
3.1.2	BATCHMAN	24
3.2	Beschreibung der Funktionsweise anhand einfacher Beispiele	24
3.2.1	Allgemeines zur Vorhersagemethode	25
3.2.2	Vorhersage einer Geraden	28
3.2.3	Vorhersage einer periodischen Funktion	37
4	Analyse und Reduktion der Zeitreihen	46
4.1	Erdrotation (Δ LOD)	46
4.1.1	Darstellung der Zeitreihe und Frequenzanalyse	46
4.1.2	Gezeitenmodellierung	46
4.1.3	Saisonale Einflüsse und linearer Anteil	51
4.1.4	Reduktion von Δ LOD um alle bekannten Einflüsse	57
4.2	Erdrotation (Δ UT1)	58
4.2.1	Darstellung der Zeitreihe und Frequenzanalyse	58
4.2.2	Gezeitenmodellierung	58
4.2.3	Saisonale Einflüsse und linearer Anteil	58
4.2.4	Reduktion von Δ UT1 um alle bekannten Einflüsse	62
4.3	Polbewegung	62
4.3.1	Darstellung der Zeitreihe und Frequenzanalyse	62
4.3.2	Saisonale Einflüsse, Chandler-Schwingung und linearer Anteil	64
4.3.3	Reduktion der Polbewegung um alle bekannten Einflüsse	70

5	Vorhersage der Erdrotationsparameter	73
5.1	Kurzfristige Schwankungen	73
5.1.1	Vorhersage der Polbewegung	73
5.1.2	Vorhersage von $\Delta UT1$	83
5.1.3	Vorhersage von ΔLOD	86
5.2	Mittelfristige Entwicklung	92
5.3	Verwendung der Zeitreihe SPACE98	92
5.4	Anwendung: Vorhersage für das Jahr 2000	95
5.5	Anwendung: Langfristige Entwicklung der Polbewegung	99
5.5.1	Testen verschiedener Konfigurationen	100
5.5.2	Trainieren des Netzes	101
5.5.3	Vorhersage	101
6	Weitere Vorhersagemethoden und Vergleich	103
6.1	Vorgehen und Qualität weiterer Vorhersagemethoden	103
6.1.1	Vorhersagemethode nach Zhu (1981, 1982)	103
6.1.2	Vorhersagemethode nach Chao (1984)	104
6.1.3	Vorhersagemethode nach McCarthy und Luzum (1990)	104
6.1.4	Vorhersagemethode nach Malkin und Skurikhina (1996)	105
6.1.5	Vorhersagegenauigkeiten nach Malkin (1999)	106
6.2	Bewertung und Einordnung des Verfahrens mit neuronalen Netzen	106
6.2.1	Polbewegung	109
6.2.2	Erdrotation	109
7	Zusammenfassung und Ausblick	110
8	Anhang: Entwickelte Software	113

Kapitel 1

Einleitung

Erdrotationsparameter

Die Entwicklung hochpräziser Raumverfahren in der physikalischen Geodäsie innerhalb der letzten zwei Jahrzehnte führte zu einer signifikanten Genauigkeitssteigerung bei der Bestimmung der Erdrotation. Vor allem die wachsenden Anforderungen der Anwender trieben diese Entwicklung voran. Dadurch wurden sowohl die Qualität bestehender Anwendungen verbessert, als auch neue Einsatzgebiete geschaffen.

Die Erdrotation kann prinzipiell mit den fünf Größen $\Delta\psi$, $\Delta\epsilon$, x , y und ΔUT1 beschrieben werden (siehe Tabelle 1.1). Die Parameter der astronomischen Nutation $\Delta\psi$ und $\Delta\epsilon$ sind nicht Gegenstand dieser Arbeit, da ihr Verlauf zu einem hohen Prozentsatz modelliert werden kann. Damit sind sie für viele Anwendungen mit ausreichender Genauigkeit vorhersagbar, insofern ist die Notwendigkeit zur Erforschung besserer Prädiktionsmethoden eher von untergeordneter Bedeutung.

Die Koordinaten x und y dienen der Beschreibung der Polbewegung, ΔUT1 ist ein Maß für den Rotationswinkel der Erde. Der Parameter ΔLOD stellt die Ableitung von ΔUT1 dar und kann somit wahlweise an dessen Stelle als 3. Parameter neben x und y gesehen werden. Auf die genauen Zusammenhänge wird in Kapitel 2.1 eingegangen.

Die aus den Genauigkeitssteigerungen erlangten Erkenntnisse über die physikalischen Zusammenhänge der Erdrotation erlauben wiederum eine exaktere Modellierung der Erdrotationsparameter.

Ihre genaue Kenntnis ist für eine große Zahl heutiger Anwendungen unerlässlich (z.B. GPS, VLBI). Bei vielen Messungen werden darüber hinaus aktuelle Werte in Echtzeit benötigt. Lasermessungen zu Satelliten oder zum Mond, terrestrische real-time Navigationsaufgaben mit GPS aber auch das Navigieren von Satelliten oder Raumfähren sind hierfür Beispiele. Da allerdings die Bestimmung bzw. Veröffentlichung der Parameter zeitverzögert erfolgt, ist es nötig, diese innerhalb des Verzögerungszeitraums zu präzisieren. Für wissenschaftliche Untersuchungen ist darüber hinaus die langfristige Entwicklung der nächsten Monate und Jahre von Interesse.

Es existieren bereits unterschiedliche Methoden für die Vorhersage, auf die später in Kapitel 6 eingegangen wird. Verschiedene Dienste, z.B. das *Subbureau for Rapid Service and Prediction* des *International Earth Orientation Service (IERS)* oder der *Earth Orientation Parameter (EOP) Service* des *Instituts für Astronomie und Astrophysik*

Parameter	Erklärung	Bedeutung der Abkürzungen
ΔLOD	Differenz der Tageslänge zu 24 Stunden	LOD: <i>Length Of Day</i> [s]
ΔUT1	UT1 – UTC	UT1: <i>Universal Time</i> (um Polbewegung korrigiert) [s] UTC: <i>Universal Time Coordinated</i> [s]
x	Polbewegungskomponente	x : X-Koordinatenwert [as]
y	Polbewegungskomponente	y : Y-Koordinatenwert [as]
$\Delta\psi$	astronomische Nutation	$\Delta\psi$: Nutation in der Länge [as]
$\Delta\epsilon$	astronomische Nutation	$\Delta\epsilon$: Nutation in der Schiefe [as]

Tabelle 1.1: Erdrotationsparameter

(IAA), bieten Vorhersagen für unterschiedliche Zeiträume an. In dieser Arbeit wird die Verwendbarkeit künstlicher¹ neuronaler Netze im Hinblick auf deren Prädiktionseigenschaften untersucht.

Zur Geschichte der neuronalen Netze

Die Geschichte der neuronalen Netze ist annähernd so alt wie die ersten programmierbaren Computer auf elektronischer Basis. Bereits Ende der 40er Jahre wurden sie zur Erkennung räumlicher Muster eingesetzt. Ihre erste Blütezeit (1955–1969) verdanken sie vor allem Frank Rosenblatt, der den ersten erfolgreichen Neurocomputer entwickelte und seine Erfahrungen und Kenntnisse in seinem 1959 erschienenen Buch *Principles of Neurodynamics* niederschrieb. Aufgrund mathematischer Analysen kamen Marvin Minsky und Seymour Papert 1969 in ihrem Werk *Perceptrons* zu dem Schluss, dass neuronale Netze viele einfache aber entscheidende Probleme gar nicht repräsentieren können. Diese Fehlinterpretation – wie sich erst später herausstellte – führte damals zu der Annahme, man befände sich in einer “Sackgasse” und hatte zur Folge, dass sich die anfängliche Euphorie schnell legte und in den darauffolgenden 15 Jahren kaum noch Mittel für dieses Forschungsgebiet zur Verfügung gestellt wurden.

Erst seit 1985 erleben neuronale Netze eine Renaissance. John Hopfields Erkenntnis, dass sich auch sehr schwierige Optimierungsaufgaben (das berühmt-berüchtigte *traveling salesman problem*) mit neuronalen Netzen lösen lassen, führte ebenso zu einer neuen Blütezeit dieser Forschungsrichtung wie das von Rummelhart, Hinton und Williams 1986 entwickelte neue Lernverfahren *Backpropagation* (siehe Kapitel 2.2.2).

Heute erfreuen sich neuronale Netze eines immer breiteren Einsatzes in Wissenschaft und Praxis. Neben Mustererkennung, Klassifikation und Steuerung von Maschinen ist auch die Vorhersage von Zeitreihen ein mögliches Einsatzgebiet.

Ein kurzer Überblick über die Arbeit

Die Ausgangsinformation und Grundlage für die folgende Arbeit stellen die Erdrotationsparameter in Form von Zeitreihen dar. Diese liegen aus unterschiedlichen Quellen in verschiedenen Auflösungen und Genauigkeiten vor. Beispiele hierfür sind die C01- und C04-Reihe des IERS, die SPACE-Reihe (Gross) und die neue aus Reanalysen astronomischer Beobachtungen unter Benutzung des HIPPARCOS-Kataloges gewonnene Reihe (Vondrák).

Erste Versuche, die Erdrotationsparameter mit Hilfe neuronaler Netze vorherzusagen, sind bereits unternommen worden (Egger, 1992; Egger und Fröhlich, 1992; Moll, 1999). In dieser Arbeit soll ein neuer Weg beleuchtet werden: Der erste Schritt besteht darin, alle bekannten Effekte, die Einfluss auf die Erdrotationsparameter nehmen, zu modellieren. Im einzelnen sind dies die Gezeiten der festen Erde, die Meeresgezeiten und die saisonalen atmosphärischen Variationen. Nur die Differenzen zwischen den modellierten und den beobachteten Parametern, d.h. quasiperiodische und unregelmäßige Variationen, dienen als Grundlage für die Vorhersage.

Zum Konstruieren und Validieren der neuronalen Netze wird der *Stuttgarter Neuronale Netze Simulator (SNNS)* angewandt. Dieser stellt ein sehr leistungsstarkes Software-Paket dar und ist darüber hinaus sowohl im Internet als auch in mehreren Linux-Distributionen als freie Software verfügbar. Mit seiner Hilfe soll der optimale Netzwerktyp, die ökonomischste Netzwerkarchitektur, das schnellste Lernverfahren und die beste Methode, das Netz mit Datenmustern zu füttern, gefunden werden.

¹das Adjektiv *künstlich* wird im folgenden nur noch dann vorangestellt, wenn die Gefahr einer Verwechslung mit dem biologischen neuronalen Netz besteht

Kapitel 2

Grundlagen

2.1 Erdrotationsparameter

2.1.1 Zeitskalen und Definitionen

Zeitskalen

In dieser Arbeit wird von verschiedenen Zeitskalen Gebrauch gemacht. Es ist also wichtig, sich im Vorfeld mit deren Eigenschaften auseinanderzusetzen. Sehr wichtig ist auch die Kenntnis über Transformationsmöglichkeiten zwischen den verschiedenen Skalen.

Wir können die verwendeten Zeitskalen zunächst grob nach ihrer Entstehung bzw. Erzeugung in natürliche und statistische unterteilen. Natürlich bedeutet, dass diese Skalen aus periodischen Vorgängen in der Natur (z.B. Erdrotation, Umlauf der Erde um die Sonne) abgeleitet werden und damit beobachtbar sind. Der Atomzeitskala liegt dagegen ein statistischer Prozess (Definition der SI-Sekunde) zugrunde. In Tabelle 2.1 sind alle in dieser Arbeit auftretenden Zeitskalen beschrieben (Müller, 1999). UT0 ist gleichzusetzen mit der mittleren Sonnenzeit bezogen auf den Nullmeridian von Greenwich. Da die erreichbare Genauigkeit bei der Messung der Sonnenposition allerdings viel zu gering ist, wird UT0 mit Hilfe einer Formel aus der präziser messbaren Sternzeit abgeleitet. Bei dieser Umrechnung wird angenommen, dass die Differenz der geographischen Längen zwischen Beobachtungsort und dem Nullmeridian konstant bleibt. Da dies jedoch aufgrund der Polschwankungen nicht der Fall ist, muss UT0 eben um diese Polschwankung korrigiert werden. Die damit erhaltene Zeit UT1 ist jetzt zwar für alle Beobachtungsorte der Erde konsistent, wegen der Variation der Erdrotationsgeschwindigkeit ist sie allerdings immer noch ungleichförmig. Eine Korrektur von UT1 um die jahreszeitlichen Schwankungen liefert die Zeit UT2. Auf die Zeitskalen UT1R, UT1S und UT1D wird in Kapitel 4 näher eingegangen.

Der deutlich genauere und leichter zugängliche Zeitstandard TAI wird mit Hilfe der SI-Sekunde durch Atomuhren realisiert. Die Definition von TAI ist der gewichtete Mittelwert der Messungen vieler über die Erde verstreuter Labors. Da die gemessene Zeit nach der speziellen Relativitätstheorie von der Höhe und der Geschwindigkeit abhängt, bezieht sich TAI auf einen Messort, der sich mit der Erde mitdreht und auf mittlerer Meereshöhe liegt. Der Ursprung von TAI wurde so festgelegt, dass am 1. Januar 1958 $UT1 - TAI = 0$ ist. Die Unsicherheit von TAI ist im Vergleich zu UT1 ungefähr 6 Größenordnungen geringer.

Die Einheit der Coordinated Universal Time (UTC) ist ebenfalls die SI-Sekunde. Der Unterschied zu TAI ist, dass die Differenz zu UT1 dem Betrag nach nicht größer als 0.9 Sekunden sein soll. Da die SI-Sekunde nicht mit der UT1-Sekunde übereinstimmt, driftet UTC gegenüber UT1. Um die oben genannte Bedingung bezüglich der Differenz $UTC - UT1$ einzuhalten, werden vom IERS gelegentlich Schaltsekunden eingefügt. Damit verbindet UTC die Vorteile von UT1 und UTC: Eine hochkonstante Zeiteinheit und möglichst gute Übereinstimmung mit dem Sonnenlauf (Husfeld, 1996).

Das Julianische Datum (JD) ist eine fortlaufende Zählung der Tage, startend mit dem Tag 0, der am 1. Januar 4713 v. Chr. um 12 Uhr mittags begann. Die Nachkommastellen beschreiben dabei die Tageszeit. Das Julianische Datum 2 451 544.5 bezeichnet also den 1. Januar 2000 um 0 Uhr. JD kann mit Hilfe einer Formel aus UT1 errechnet werden. Da beim JD sehr große Zahlenwerte auftreten, wird häufig das Modifizierte Julianische Datum (MJD) verwendet. Von JD wird dabei einfach ein konstanter Wert subtrahiert:

$$MJD = JD - 2\,400\,000.5 \tag{2.1}$$

	Zeitskala	Beschreibung	Entstehung
Natürlich	UT0	Universal Time (Weltzeit) entspricht der mittleren Sonnenzeit	astronomische Beobachtungen
	UT1	Um Einfluss der Polbewegung korrigiertes UT0	Reduktion von UT0
	UT2	Um Einfluss der jahreszeitlichen Schwankungen korrigiertes UT1	Reduktion von UT1
	UT1R	UT1 um kurzperiodischen Anteil der Erdzeiten (5 bis 35 Tage) reduziert	Reduktion von UT1
	UT1S	UT1 um Erdzeiten (5 Tage bis 18.6 Jahre) und Meereszeiten (> 1 Tag) reduziert	Reduktion von UT1
	UT1D	UT1 um kurzperiodischen Anteil der Meereszeiten (≤ 1 Tag) reduziert	Reduktion von UT1
Statistisch	TAI	Temps Atomique International (internationale Atomzeit)	SI-Sekunde, Atomuhren
	UTC	Universal Time Coordinated (koordinierte Weltzeit)	TAI durch Schaltsekunden an UT1 angepasst
	JD	das Julianische Datum ist definiert durch die fortlaufende Zählung der Tage seit 1.1.4713 v. Chr. 12 Uhr	fortlaufende Zählung von UT
	t	Julianische Jahrhunderte	$t = (JD - 2451545) / 36525$

Tabelle 2.1: Definition der verwendeten Zeitskalen

Der Nullpunkt liegt deshalb beim 17. November 1858 um 0 Uhr, und Sylvester 2000 entspricht somit einem MJD von 51 544.

Oft wird die Zeit auch in der Einheit *Julianische Jahrhunderte* benötigt. Diese gibt die vergangenen Jahrhunderte seit dem 1.1.2000 um 0 Uhr in Dezimalschreibweise an, d.h. für das 20. Jahrhunderts besitzt t ein negatives Vorzeichen.

$\Delta UT1$

$\Delta UT1$ ist die beobachtete Differenz der Erdrotationsphase gegenüber UTC:

$$\Delta UT1 = UT1 - UTC \tag{2.2}$$

In $\Delta UT1$ enthalten sind damit sowohl die Erdrotationsschwankungen aufgrund zonaler Gezeiten, atmosphärischer Zirkulation oder Effekte aus dem Erdinneren als auch die vom IERS eingefügten Schaltsekunden gegenüber einer gleichförmigen Zeitskala. Vor dem Jahr 1972 erfolgte die Anpassung von UTC an UT1 nicht durch Einfügen konstanter Sekundensprünge, sondern durch einen linearen Term, um Unstetigkeiten zu vermeiden. Abbildung 2.1 zeigt u.a. die gemessene Zeitreihe $\Delta UT1 = UT1 - UTC$. Sehr schön zu erkennen ist die Drift von UT1 gegenüber TAI. Die Ursache für dieses Phänomen ist allerdings in erster Linie nicht die sich immer langsamer drehende Erde – dies macht weniger als 2 ms in 100 Jahren aus –, sondern die Tatsache, dass zur Festlegung der SI-Sekunde das Jahr 1900 herangezogen wurde, das kürzer war als die durchschnittliche Jahreslänge (Husfeld, 1996). D.h. die definierte SI-Sekunde ist kürzer als eine durchschnittliche UT1-Sekunde, was dazu führt, dass zu (fast) jedem beliebigen Zeitpunkt mehr SI-Sekunden als UT1-Sekunden bezogen auf 1900 vergangen sind.

ΔLOD

Die Differenz zwischen astronomisch bestimmter Tageslänge *length of day* (LOD) zu 86 400 s der Atomzeitskala wird als ΔLOD bezeichnet und meist in Millisekunden (ms) angegeben:

$$\Delta LOD = LOD - 86\,400\,s \tag{2.3}$$

LOD steht im direkten Zusammenhang mit der Winkelgeschwindigkeit ω der Erdrotation:

$$\omega = 72\,921\,151.467064 - 0.843994803\,LOD$$

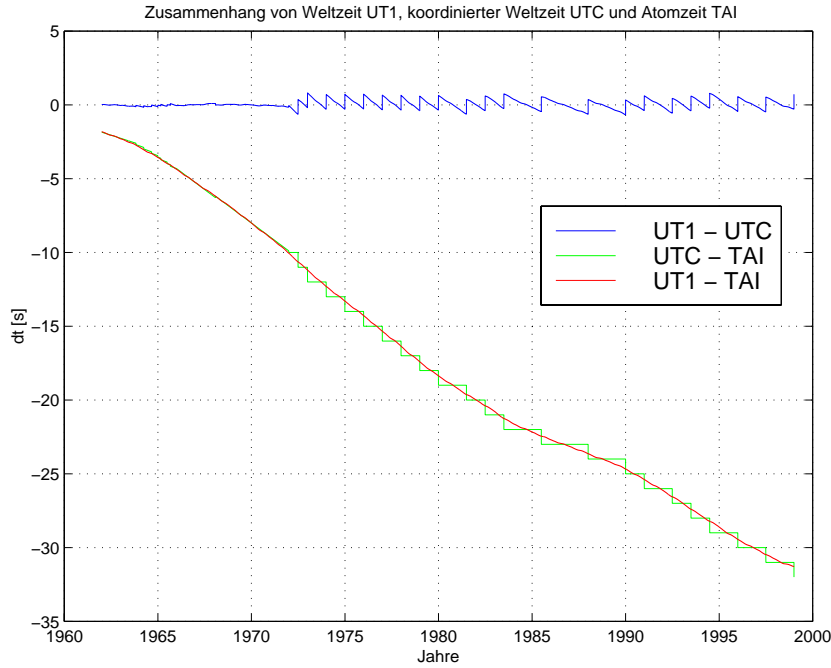


Abbildung 2.1: Zeitskalen

Dabei sind LOD in Millisekunden und ω in Picoradian/s einzusetzen.

ΔLOD ist aus $\Delta UT1$ ableitbar (Freedman et al., 1994):

$$\Delta LOD = -LOD_0 \frac{dUT1_A}{dt} \tag{2.4}$$

Hier ist LOD_0 die nominelle Tageslänge von 86 400 s und $\frac{dUT1_A}{dt}$ die Ableitung der Differenz $UT1 - TAI$ nach der Zeit. Diese Formel soll kurz für den in der Praxis vorliegenden diskreten Fall anhand eines Beispiels erläutert werden: Liegen an zwei aufeinanderfolgenden Tagen die Werte für $\Delta UT1$ im Abstand von 24 Stunden vor, so können wir zunächst durch Korrektur der Schaltsekunden $dUT1_A$ für beide Epochen berechnen. Realistische Werte sind z.B. für $dUT1_A^1 = -1.813 s$ und für $dUT1_A^2 = -1.815 s$. Dann ist

$$\begin{aligned} \Delta LOD &= -86\,400 s \frac{dUT1_A^2 - dUT1_A^1}{\Delta t_{1,2}} \Rightarrow \\ \Delta LOD &= -86\,400 s \frac{-1.815 s - (-1.813 s)}{24 \cdot 60 \cdot 60 s} \\ &= 2 \text{ ms} \end{aligned} \tag{2.5}$$

ΔLOD schwankt ungefähr zwischen -0.5 und 4.5 ms, was bedeutet, dass die meisten Tage länger als 86 400 s sind. Dieser Umstand ist wieder auf die oben genannte Definition der SI-Sekunde zurückzuführen.

Polbewegung

Die Polkoordinaten x und y beschreiben die Lage des zälestischen Ephemeridenpols (CEP) relativ zum IERS Referenzpol (IRP) im erdfesten System¹ (IERS, 1999). Sie werden in Bogensekunden (as) angegeben und liegen in einem Wertebereich, der sowohl in x - als auch in y - Richtung ca. $0.6''$ überdeckt, was im metrischen Maß einer Schwankung von ca. ± 10 m auf der Erdoberfläche entspricht.

2.1.2 Geophysikalische Zusammenhänge

Die Variation der Erdrotation wird durch verschiedene im Erdinnern ablaufende Prozesse sowie durch eine Vielzahl von Kräften hervorgerufen, die auf den Erdkörper einwirken (siehe Abbildung 2.2). Durch diese kommt es zu Massenverlagerungen innerhalb der Erde, den Ozeanen und der Atmosphäre, die zu einer Veränderung des Trägheitstensors führen.

¹Die Grundlagen der Referenzsysteme sind in Kapitel 2.1.3 dargestellt.

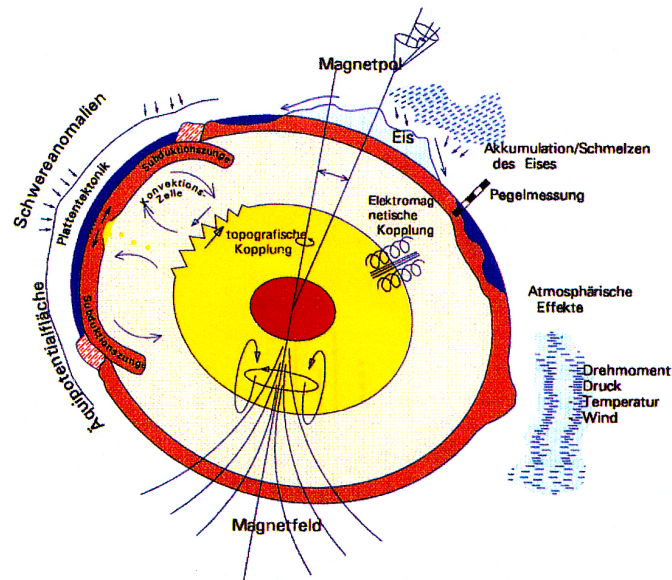


Abbildung 2.2: Schematische Darstellung der wichtigsten Kräfte, die sich auf die Erdrotation auswirken (Ballani et al., 1993)

Kurzperiodische Komponenten

- Diejenigen Prozesse, die kurzperiodische Massenumverteilungen (Perioden bis zu 10 Jahren) bedingen, geschehen hauptsächlich in der Atmosphäre und der Hydrosphäre. Durch sie können die saisonalen Variationen fast vollständig beschrieben werden. Diese sind vor allem durch die Inklination der Erde bedingt. So verändert sich der Einstrahlwinkel der Sonne innerhalb eines Jahres. Die dadurch eintretende wechselnde Erwärmung und Abkühlung der Hemisphären bewirkt eine Massenverlagerung. Hiermit sind also die jährlichen Schwankungen der Erdrotation zu erklären (Ballani et al., 1993).

Neben diesen treten allerdings auch Veränderungen des Rotationsvektors mit Perioden eines halben Jahres auf. Diese Erscheinung können wir uns mit der unterschiedlichen Aufteilung der Kontinente auf Süd- und Nordhemisphäre begreiflich machen. Auf der Nordhalbkugel finden wir mehr Kontinente, dafür weniger "Wasserfläche" als auf der Südhalbkugel. Daher wirken sich die Unterschiede in der Sonneneinstrahlung auf Süd- und Nordhalbkugel verschieden aus, was sich in einer auffälligen halbjährlichen Komponente im Rotationspektrum bemerkbar macht.

Die Polbewegung besitzt neben den saisonalen Komponenten noch die sogenannte Chandler-Periode. Diese ist mit einer frei rotierenden, deformierbaren (elastischen) Erde erklärbar. Da hierbei kein Drehmoment von außen wirkt, wird sie auch als freie Polbewegung bezeichnet. Sie besitzt eine Periode von ca. 435 Tagen. Würden wir von einer starren Erde ausgehen, entspräche diese Schwingung der bekannten Euler-Periode von ca. 305 Tagen (siehe Drewes, 1998).

- Aber auch die Gezeiten der festen Erde und die Ozeangezeiten sind Verursacher kurzperiodischer Schwankungen. Die Gezeitenbeschleunigung, die durch eine Masse M im Abstand r von der Erde verursacht wird, können wir mit $\frac{GM}{r^2} \cdot \frac{R}{r}$ abschätzen. R ist dabei der Erdradius (Brosche und Schuh, 1998). Die exakte allgemeine Formel für den Einfluss von Sonne oder Mond auf die Gezeitenbeschleunigung lautet:

$$\ddot{\vec{r}} = GM \left(\frac{\vec{r}_{s,m} - \vec{r}}{|\vec{r}_{s,m} - \vec{r}|^3} - \frac{\vec{r}_{s,m}}{r_{s,m}^3} \right) \quad (2.6)$$

Dabei ist

- $\ddot{\vec{r}}$ der resultierende Beschleunigungsvektor im Berechnungspunkt,
- \vec{r} der geozentrische Ortsvektor des Berechnungspunktes,
- $\vec{r}_{s,m}$ der geozentrische Ortsvektor von Sonne oder Mond und
- $r_{s,m}$ der Betrag von $\vec{r}_{s,m}$.

Da die Ephemeriden von Sonne und Mond im geozentrischen System sehr genau berechenbar sind, können wir für jeden beliebigen Orts- und Zeitpunkt die Gezeitenbeschleunigung errechnen. Die Darstellung der Überlagerung der Effekte von Sonne und Mond geschieht mit Hilfe der Kugelflächenfunktionen (Rummel, 1992).

Wir können die Gezeiten gliedern in zonale, sektorielle und tesserale. Die zonalen Gezeiten sind nur abhängig von der geographischen Breite, wohingegen die sektoriellen lediglich eine Funktion der geographischen Länge darstellen (siehe Abbildungen 2.3 (a) und (b)). Tesserales Gezeiten besitzen beide Eigenschaften (siehe Abbildung 2.3 (c)). Zonale Gezeiten sind also rotationssymmetrisch und verändern durch die Verlagerung der Massen entlang des Meridians das Trägheitsmoment bezüglich der Rotationsachse (in Richtung der Pole nimmt der Abstand zur Rotationsachse ab). Dies führt zu einer Veränderung der Rotationsgeschwindigkeit der Erde. Nachvollziehen können wir diese Erscheinung besser, wenn wir uns z.B. eine Eiskunstläuferin vorstellen. Um eine schnellere Drehbewegung zu erreichen, legt sie ihre Arme eng an den Körper. Möchte sie ihre Drehbewegung verlangsamen, streckt sie ihre Arme wieder aus. Dieser Effekt ist vergleichbar mit den Auswirkungen zonaler Gezeiten auf UT1 bzw. LOD.

Da der Erdkern flüssig ist, deformiert er sich beim Einwirken des Gravitationspotentials von Sonne und Mond mehr als der Mantel. Dadurch sind die Rotationen von Kern und Mantel voneinander losgelöst, was zu der Bezeichnung *entkoppelter Kern* führte. Diese Tatsache und sowohl die Ozeangezeiten selbst als auch die Verstärkung der Erdgezeiten durch die Ozeangezeiten berücksichtigen Yoder et al. (1981) in ihrem Modell. Dies erreichen sie, indem sie die *Lovesche Zahl* k , die das Ausmaß und die Geschwindigkeit der Deformation der Niveauebenen aufgrund der Gezeiten beschreibt, durch eine *effektive Lovesche Zahl* ersetzen, die den oben genannten Effekt berücksichtigt. Mit diesem Modell können wir nun aus den Gezeitendeformationen durch Sonne und Mond die periodischen Variationen der Erdrotation ermitteln. Es umfasst Gezeiten mit Perioden von 5 Tagen bis zu 18.6 Jahren. Seine Anwendung wird in Kapitel 4.1.2 beschrieben.

Sektorielle und tesserale Gezeiten haben dagegen keinen merklichen Einfluss auf die Rotationsgeschwindigkeit der Erde. Auf die Polbewegung haben weder zonale noch sektorielle Gezeiten Einfluss, da das Trägheitsmoment bezüglich der 2. und 3. Hauptträgheitsachse (liegen in der Äquatorebene) nicht beeinflusst wird.

Neben einer Verstärkung der Erdgezeiten führen die Ozeangezeiten auch direkt zu Variationen der Erdrotation. Im oben genannten Spektralbereich von 5 Tagen bis 18.6 Jahren ist dieser Einfluss allerdings sehr viel kleiner als der der Erdgezeiten selbst und kann in vielen Fällen vernachlässigt werden. Im hochfrequenten Bereich mit Perioden von einem Tag und weniger sind sie hingegen unbedingt zu berücksichtigen. Die Ozeangezeiten wirken sich auch auf die Polbewegung aus, da die Ozeane asymmetrisch zur Rotationsachse über die Erdoberfläche verteilt sind (Ballani et al., 1993; Brosche und Schuh, 1998).

Langperiodische Komponenten

Vorgänge in den festen Teilen des Erdkörpers, vor allem im Kern-Mantel-Bereich, rufen langperiodische Veränderungen (Perioden größer als 10 Jahre) hervor. Es besteht die Vermutung, dass diese sogenannten dekadischen Fluktuationen der Erdrotation entweder durch Prozesse entstehen, die die Massengeometrie der Gesamterde relativ zur Rotationsachse verändern oder solche, die in Bewegung nahe der Kern-Mantel-Grenze bestehen und mit der Mantelrotation gekoppelt sind (Ballani, 1993). Auf diesem Gebiet ist allerdings noch weitere Forschungsarbeit notwendig.

Auch Klimaveränderungen wirken sich indirekt auf die Erdrotation aus, indem sie die Polkappen abschmelzen lassen und so zu einer Massenverlagerung von den Eisflächen hin zu den Weltmeeren führen.

2.1.3 Anwendung der Erdrotationsparameter

Die Erdrotationsparameter beschreiben die Rotation des vereinbarten terrestrischen Systems (*Conventional Terrestrial System, CTS*), das beispielsweise vom IERS mit dem *International Terrestrial Reference Frame (ITRF)* umgesetzt wurde, zum vereinbarten Inertialsystem (*Conventional Inertial System, CIS*), das vom IERS wiederum durch das *International Celestial Reference Frame (ICRF)* realisiert wurde, in Verbindung mit dem vereinbarten Präzessions-Nutations-Modell (Zebhauser, 1999). Die Transformation eines Punktes vom inertialen zum terrestrischen System kann durch die Gleichung

$$x_T = W(t) S(t) N(t) P(t) x_I \quad (2.7)$$

beschrieben werden (McCarthy, 1996). Die Bedeutung der einzelnen Rotationsmatrizen wird im Folgenden kurz erläutert:

Kugelflächenfunktionen

Abbildung 2.3: Kugelflächenfunktionen

Ausgangssystem ist das vereinbarte raumfeste Äquatorsystem (hier: ICRF) zur Epoche t_0 . Die z-Achse dieses mittleren Äquatorsystems entspricht dem mittleren Himmelspol zur Epoche t_0 .

$P(t) = R_3(-z) R_2(\theta) R_3(-\zeta)$: Rotationsmatrix zur Reduktion der Präzession.

Hierbei können die drei Größen z , θ und ζ als Funktion der Zeit für jeden beliebigen Zeitpunkt sehr genau berechnet werden.

$N(t) = R_1(-\epsilon - \Delta\epsilon) R_3(-\Delta\psi) R_1(\epsilon)$: Rotationsmatrix zur Reduktion der Nutation.

ϵ ist die mittlere Schiefe der Ekliptik und dabei eine Funktion der Zeit, $\Delta\psi$ als astronomische Nutation in der Länge und $\Delta\epsilon$ als astronomische Nutation in der Schiefe sind Funktionen der Parameter der Mondbahn D , F und Ω , die ebenso für jeden Zeitpunkt bekannt sind. Leider ist dieses Modell für praktische Zwecke oft nicht ausreichend. Die Unterschiede zwischen den berechneten und beobachteten Größen werden als Zuschläge $\delta\Delta\psi$ und $\delta\Delta\epsilon$ unter der Bezeichnung **celestial pole offset** (*Himmelspol-Offset*) vom IERS veröffentlicht. Diese Korrekturen sind Bestandteil der Erdrotationsparameter. Die korrigierten Nutationsparameter ergeben sich dann zu

$$\Delta\psi = \Delta\psi_{\text{altes Modell}} + \delta\Delta\psi$$

und

$$\Delta\epsilon = \Delta\epsilon_{\text{altes Modell}} + \delta\Delta\epsilon.$$

Wie oben schon erwähnt, ist die Vorhersage von $\Delta\psi$ und $\Delta\epsilon$ weitaus weniger kritisch.

Nach Anbringen der Reduktionen für Präzession und Nutation befinden wir uns in einem momentanen Äquatorsystem, in dem die z-Achse dem CEP entspricht. Der CEP ist annähernd mit der tatsächlichen Rotationsachse der Erde gleichzusetzen. Die Differenz zwischen beiden beträgt allerdings max. $0.01''$. Der Übergang zum vereinbarten terrestrischen System *conventional terrestrial system* (CTS, hier: ITRF) erfolgt durch Berücksichtigung des Erdrotationswinkels und der Polbewegung.

$R(t) = R_3(GST)$: Rotationsmatrix zur Reduktion des Erdrotationswinkels.

Greenwich Sidereal Time (GST) ist hauptsächlich Funktion von UT1. Da UT1 nicht direkt berechnet werden kann, wird vom bekannten UTC ausgegangen und die **Differenz von UT1 zu UTC** als Erdrotationsparameter $\Delta UT1 = UT1 - UTC$ addiert:

$$UT1 = UTC + \Delta UT1$$

$\Delta UT1$ ist aufgrund starker Schwankungen sehr schwer vorhersagbar und stellt daher hohe Ansprüche an Prädiktionsverfahren.

$W(t) = R_2(-x) R_1(-y)$: Rotationsmatrix zur Reduktion der Polbewegung.

Hier gehen nur die **Koordinaten der Polbewegung** x und y ein.

Die z-Achse dieses Systems entspricht der Koordinatenachse der im CTS festgelegten Punkte.

Zur Veranschaulichung dieser doch etwas schwer überschaubaren Theorie soll Abbildung 2.4 dienen. Hier sind die Übergänge zwischen den einzelnen Systemen noch einmal vereinfacht graphisch dargestellt.

Die Erdrotationsparameter gehen also in die Rotationsmatrizen $N(t)$, $R(t)$ und $W(t)$ ein und stellen das Bindeglied zwischen den terrestrischen Koordinaten und den Koordinaten im inertialen Raum dar. Beobachtet werden sie mit den verschiedenen geodätischen Raumverfahren GPS, VLBI, SLR/LLR oder DORIS. VLBI ist in der Lage, alle Rotationsparameter zu bestimmen. Die Satellitenverfahren erlauben keine direkte Bestimmung von $\Delta UT1$.

2.2 Neuronale Netze

2.2.1 Arbeitsweise des Gehirns

Geschwindigkeit

Im Gegensatz zum Computer, der von der Neumann-Architektur beherrscht wird, stellt das Gehirn ein gigantisches Parallelverarbeitungssystem dar, in dem Milliarden von Neuronen als Prozessoren dienen und eingehende

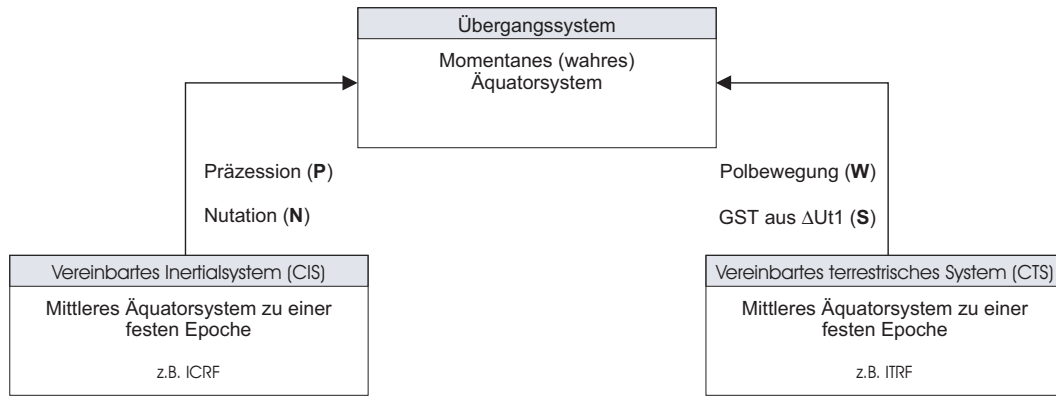


Abbildung 2.4: Erdrotationsparameter als Bindeglied zwischen den Bezugssystemen (in Anlehnung an Drewes, 1998)

	Gehirn	Rechner
Anzahl Verarbeitungselemente	ca. 10^{11} Neuronen	ca. 10^9 Transistoren
Art	massiv parallel	im allg. seriell
Speicherung	assoziativ	adressbezogen
Schaltzeit eines Elements	ca. 1 ms	ca. 1 ns
“Schaltvorgänge”/s	ca. $10^3/s$	ca. $10^9/s$
“Schaltvorgänge” insges. (theoretisch)	ca. $10^{13}/s$	ca. $10^{18}/s$
“Schaltvorgänge” insges. (tatsächlich)	ca. $10^{12}/s$	ca. $10^{10}/s$

Tabelle 2.2: Vergleich zwischen Gehirn und Rechner

Reize verarbeiten. Vergleichen wir allerdings die Verarbeitungsgeschwindigkeit des Gehirns mit der eines modernen Computers, so stellen wir fest, dass der Computer dem menschlichen Gehirn um 6 Zehnerpotenzen überlegen ist. Tabelle 2.2 aus Zell (1994) bietet eine sehr interessante Gegenüberstellung. Auffällig sind hier vor allem zwei Tatsachen: Die beim Rechner - wie bereits festgestellt- wesentlich größere Anzahl an Schaltvorgängen pro Sekunde und das beim Gehirn deutlich größere Verhältnis zwischen der tatsächlichen und theoretischen Leistung. Die Ursache hierfür liegt in der hochgradig parallelen Verarbeitung des Gehirns. So ist in der menschlichen Schaltzentrale zu jedem Zeitpunkt ein hoher Prozentsatz der Neuronen aktiv, während beim Rechner nur der Prozessor, nicht aber die Transistoren permanent arbeiten. Allzu ernst sollten wir diese Gegenüberstellung allerdings nicht nehmen, da Neuronen nur sehr bedingt mit Transistoren vergleichbar sind. Trotzdem vermag sie doch, einen guten Eindruck von der hohen Gesamtleistung des menschlichen Gehirns und dessen Ursache zu vermitteln. Veranschaulichen lässt sich dies auch sehr schön durch die sog. *100-Schritt-Regel*: Während der Mensch ca. 0.1 s, d.h. 100 *sequentielle* Einzelschritte bei einer Schaltzeit von 1ms, benötigt, um einen bekannten Gegenstand zu erkennen, kann ein Rechner in nur 100 Verarbeitungsschritten kaum etwas erreichen (Zell, 1994).

Biologische Zusammenhänge

Um die Ansätze im Zusammenhang mit künstlichen neuronalen Netzen zu verstehen, erscheint es wichtig, die Materie zunächst aus biologischer Sicht zu beleuchten. Dies kann und soll im Rahmen der Arbeit nur in stark vereinfachter Form geschehen.

Die Hirnrinde (*Neokortex*) ist Sitz der intelligenten Leistung des Gehirns. Diese $0.2 m^2$ grosse und ca. 2–3mm dicke Schicht besteht aus Nervenzellengewebe (Ritter et al., 1990). Auf einem Quadratmillimeter der Hirnrinde befinden sich etwa 100 000 vernetzte Nervenzellen (Neuronen). Jedes dieser insgesamt ca. 10^{11} Neuronen ist durchschnittlich mit 10 000 Nachbarneuronen verbunden. Bereits bei der Geburt sind *sämtliche* Neuronen ausgebildet. Um sich deren große Anzahl zu verdeutlichen, können wir uns vorstellen, dass beim Menschen während der Schwangerschaft jede Sekunde durchschnittlich 4 000 Neuronen entstehen müssen, um bei der Geburt auf dieses Soll zu kommen (Zell, 1994).

Die Vernetzung beginnt dagegen erst nach der Geburt mit dem Lernen. Optimale Reaktionen auf bestimmte Herausforderungen der Umwelt speichert das Neugeborene durch Verbindungsmuster zwischen den Neuronen. Das

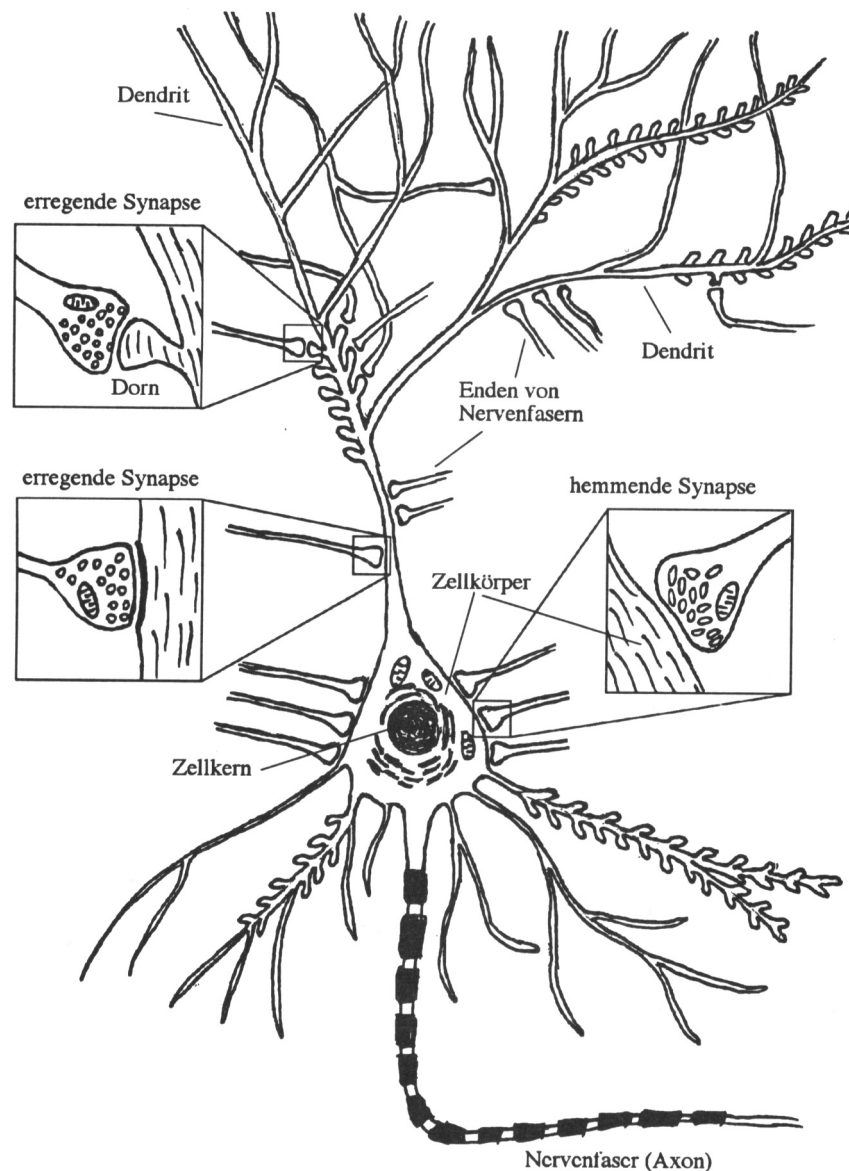


Abbildung 2.5: Das Neuron als biologisches Vorbild (Zell, 1994)

Trainieren bestimmter Verhaltensformen stärkt die entsprechenden Verbindungen (Lernen), Vernachlässigung dagegen bewirkt deren Zerfall (Vergessen).

Die eigentlichen Verarbeitungseinheiten im Gehirn sind die Neuronen, die wie jeder Prozessor Eingabe und Ausgabe besitzen (siehe Abbildung 2.5).

Die Eingabesignale sind elektrochemische Reize, die von benachbarten Neuronen über spezielle Leitungen, den *Dendriten*, an das Neuron übermittelt werden. Die Menge aller in ein Neuron mündenden Dendriten wird als *Dendritenbaum* bezeichnet. Dabei wird dem Zellkörper die Summe aller eingehenden elektrochemischen Reize als elektrisches Potential zugeleitet. Überschreitet dieses Potential einen gewissen Schwellwert, wird das Neuron aktiv und sendet über eine Ausgabeleitung, dem *Axon*, einen kurzen elektrischen Nadelimpuls. Dieses Axon ist ebenfalls verzweigt und führt dadurch den Impuls an bis zu mehrere tausend Zielneuronen weiter. Im Zielgebiet bildet das Axon an den Enden der Verästelung Verdickungen aus, die sogenannten *Synapsen*. Diese Zellen befinden sich entweder direkt auf dem Zellkörper des Zielneurons oder auf dem Dendritenbaum. Sie sind in der Lage, den ankommenden Potentialwert auf chemischem Wege zu verstärken (exzitatorische Synapse) oder zu hemmen (inhibitorische Synapse). Dadurch, dass sie ihre verstärkende oder hemmende Wirkung verändern können, spielen sie im bereits erwähnten Lernprozess die entscheidende Rolle.

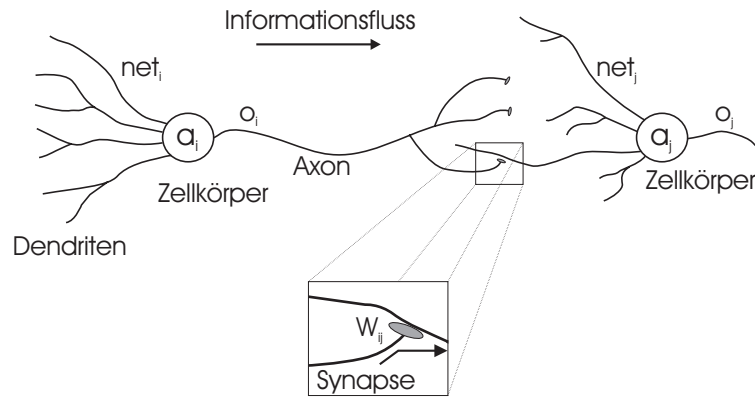


Abbildung 2.6: Neuronales Netz - Idealisierungsstufe I

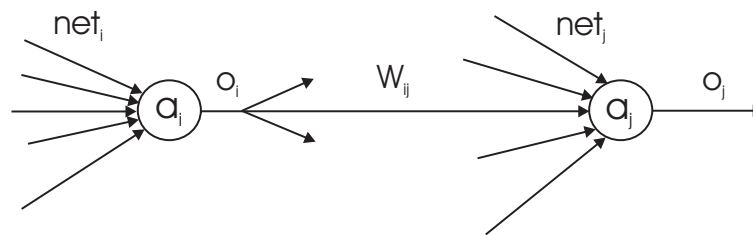


Abbildung 2.7: Neuronales Netz - Idealisierungsstufe II

Detailliertere Schilderungen der biologischen Zusammenhänge lassen sich unter anderem in Zell (1994), Ritter et al. (1990) oder natürlich in der Biologieliteratur nachlesen.

2.2.2 Mathematische Darstellung

Der Übergang vom biologischen Vorbild zum mathematischen Modell ist weitaus weniger komplex, als viele wahrscheinlich zunächst vermuten würden, würden sie mit dem Begriff "neuronales Netz" oder "künstliche Intelligenz" konfrontiert werden. Das im Abschnitt 2.2.1 beschriebene biologische Modell muss lediglich so formuliert werden, dass es für Computersimulationen geeignet ist und durch mathematische Formalismen fassbar wird.

Korrespondenz zwischen einzelnen Zellen

Für diese Aufgabe erscheint es hilfreich, die in Abbildung 2.5 bereits vereinfachte Darstellung zu idealisieren. Die Zellen (Neuronen) eines künstlichen neuronalen Netzes bestehen wie auch die Neuronen des biologischen Vorbilds aus drei Komponenten: einem Zellkörper, den Dendriten und einem Axon. Dabei verzweigt sich das Axon, das die Ausgabe eines Zellkörpers weiterleitet und tritt mit den Dendriten nachfolgender Neuronen über die Synapsen in Kontakt (siehe Abbildung 2.6, in Anlehnung an Zell, 1994). Die verstärkende bzw. hemmende Wirkung der Synapsen zwischen zwei Neuronen a_i und a_j wird durch ein positives bzw. negatives Verbindungsgewicht w_{ij} dargestellt. Indem wir einen zweiten Idealisierungsschritt anschließen, können wir nun in der graphischen Darstellung die Verbindung zwischen Neuronen als direkte gewichtete Verbindung zwischen den Zellkernen a_i und a_j darstellen (siehe Abbildung 2.7 in Anlehnung an Zell, 1994).

Erläuterungen zur Abbildung 2.7 und Vorgehen bei der Berechnung:

1. Die Netzeingabe (*net input*) net_j von Zelle j berechnet sich aus der Summe aller Ausgaben (*net output*) der Vorgängerezellen multipliziert mit den entsprechenden Gewichten der Verbindung von Zelle i nach Zelle j :

$$net_j(t) = \sum_i o_{i(t)} w_{ij} \tag{2.8}$$

2. Der sogenannte *Aktivierungszustand* $a_j(t+1)$ einer Zelle errechnet sich anschließend als Funktion der Netzeingabe $net_j(t)$, eines Schwellwertes θ_j und der alten Zellaktivierung $a_j(t)$:

$$a_j(t+1) = f_{act}(a_j(t), net_j(t), \theta_j) \tag{2.9}$$

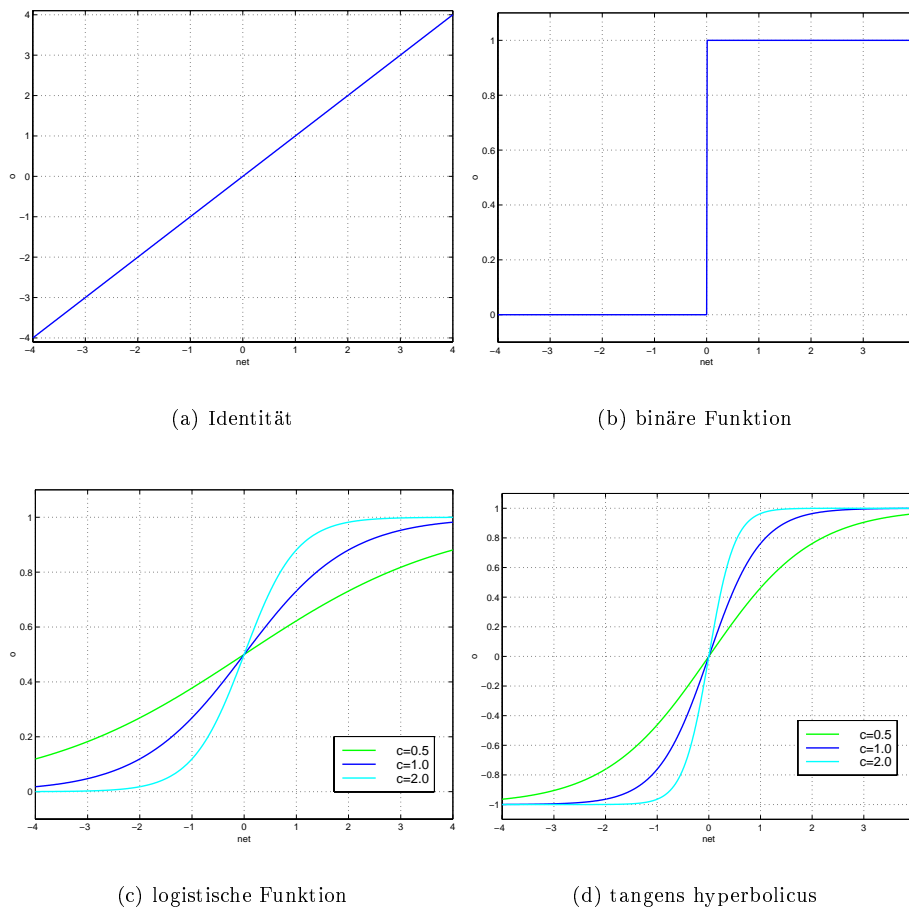


Abbildung 2.8: Typische Aktivierungs- bzw. Ausgabefunktionen

Die Funktion f_{act} wird dabei als *Aktivierungsfunktion (Transferfunktion)* bezeichnet.

- Im letzten Schritt muss nun noch die eigentliche Ausgabe o_j ermittelt werden. Diese ist wiederum eine Funktion der Aktivierung der Zelle:

$$o_j = f_{out}(a_j), \tag{2.10}$$

wobei f_{out} die sogenannte *Ausgabefunktion* darstellt

Durch Wahl einer geeigneten Aktivierungs- bzw. Ausgabefunktion können nichtlineare Zusammenhänge repräsentiert werden. Deshalb wird als Aktivierungsfunktion meist eine nichtlineare Funktion gewählt. Als Ausgabefunktion kann dann die Identitätsfunktion (d.h. $o_j = a_j$) gewählt werden, da bereits in der Aktivierungsfunktion die gewünschte Nichtlinearität enthalten ist. Bei manchen Anwendungen kann es jedoch durchaus sinnvoll sein, eine andere Ausgabefunktion zu verwenden, z.B. wenn nur das Neuron mit der höchsten Aktivierung in seiner Schicht ein Signal senden soll (*winner take all*). Einige typische Funktionen sind:

Funktionsname	math. Darstellung	graph. Darstellung
Identität	$f_{act/out}(net) = net$	siehe Abb. 2.8 (a)
binäre Funktion	$f_{act/out}(net) = \begin{cases} 0 & ; net \leq 0 \\ 1 & ; net > 0 \end{cases}$	siehe Abb. 2.8 (b)
logistische Funktion	$f_{act/out}(net) = \frac{1}{1+e^{-c \cdot net}}$	siehe Abb. 2.8 (c)
tangens hyperbolicus	$f_{act/out}(net) = \tanh(c \cdot net)$	siehe Abb. 2.8 (d)

Verwenden wir neben einer linearen Aktivierungsfunktion auch eine lineare Ausgabefunktion (z.B Identität), so

stellt jede Netzausgabe eine Linearkombination der Netzeingaben dar.²

Sollen die möglichen Aktivierungszustände auf 0 und 1 beschränkt bleiben, d.h. das Neuron sendet einen Impuls oder aber auch nicht, so verwenden wir die binäre Funktion. Möchten wir dagegen einen “weicheren” Übergang, so können wir eine sog. *sigmoide Funktion* einsetzen, deren Verlauf an ein “S” erinnert. Vertreter der sigmoiden Funktionen sind z.B. die *logistische Funktion* und der *tangens hyperbolicus*. Diese Funktionen haben den Vorteil, dass sie keine Unstetigkeitsstelle besitzen und deshalb differenzierbar sind. Außerdem erhalten wir dadurch unendlich viele mögliche Aktivierungszustände in einem endlichen Intervall. Ein weiterer Vorteil der sigmoiden Funktionen ist, dass sie wesentlich sensibler auf Signale mit kleinen Amplituden reagieren können, da die Steigung im Ursprung am größten ist. Durch Hinzufügen eines Schwellwertes θ_j kann dieser Punkt der höchsten Sensibilität beliebig verschoben werden.

Der Schwellwert θ_j gibt also die Stelle der Aktivierungsfunktion mit der größten Steigung an. Beim biologischen Vorbild entspricht er der Reizschwelle, die überwunden werden muss, damit ein Neuron einen Impuls feuert. Er kann auf zwei unterschiedliche Weisen realisiert werden (Zell, 1994):

1. Eine Möglichkeit besteht darin, jedem Neuron explizit einen Schwellwert zuzuordnen. Während dabei die Formel 2.8

$$net_j(t) = \sum_i o_i(t) w_{ij} \quad (2.11)$$

für die Netzeingabe unverändert bleibt, spezialisiert sich die Formel 2.9 zur Berechnung des Aktivierungszustandes zu

$$a_j(t+1) = f_{act}(a_{j(t)}, net_j(t) - \theta_j). \quad (2.12)$$

2. Alternativ dazu können wir ein einzelnes zusätzliches Neuron einfügen, das einen konstanten Ausgabewert von 1 besitzt. Mit diesem sind alle übrigen Neuronen des Netzes verbunden. Die Verbindungsgewichte stellen dann die Schwellwerte der jeweiligen Neuronen dar. Der Vorteil dieser Variante ist, dass die Anpassung des Schwellwerts während der Lernphase keine Sonderrolle bei der Programmierung spielt, da dieser genauso behandelt werden kann, wie alle anderen Neuronen. In Formeln ausgedrückt ergibt sich somit für die Netzeingabe

$$net_j(t) = \left(\sum_i o_i(t) w_{ij} \right) - \theta_j \quad (2.13)$$

und den Aktivierungszustand

$$a_j(t+1) = f(a_{j(t)}, net_j(t)). \quad (2.14)$$

Nachteilig wirken sich die höhere Anzahl an Gewichten und die geringere Übersichtlichkeit einer graphischen Darstellung (zusätzliche Verbindungen zu jedem Neuron können sehr störend wirken) aus.

Verflechtung der Zellen zu einem Netz

Verknüpfen wir mehrere Neuronen untereinander, so entsteht ein künstliches Netz. Die Topologie der Netze, wie sie in dieser Arbeit verwendet werden, kann in Anlehnung an das biologische Vorbild in Schichten gegliedert werden (siehe Abbildung 2.9):

- *Eingabeschicht (input layer)*
- keine, eine oder mehrere *versteckte Schicht(en) (hidden layer)*
- *Ausgabeschicht (output layer)*

Charakteristisch für das Schichtenmodell ist, dass Neuronen verschiedener Schichten untereinander vernetzt sind, während Neuronen einer Schicht miteinander keine Verbindung eingehen.

Neuronen der Eingabeschicht werden als *Eingabeneuronen* bezeichnet. Sie dienen dem Einbringen der Eingangsinformation in das Netz und besitzen daher keine Vorgängerneuronen. Neuronen der Ausgabeschicht werden als *Ausgabeneuronen* bezeichnet. Sie geben die Ausgabe des Netzes nach außen und haben deshalb keine Folgeneuronen. Neuronen der versteckten Schichten besitzen sowohl Verbindungen zu Vorgänger- als auch zu Folgeneuronen.

²In einem mehrschichtigen Netz ist eine lineare Aktivierungsfunktion nicht sinnvoll, da dies lediglich eine Linearkombination darstellt, die auch durch eine Schicht repräsentiert werden kann.

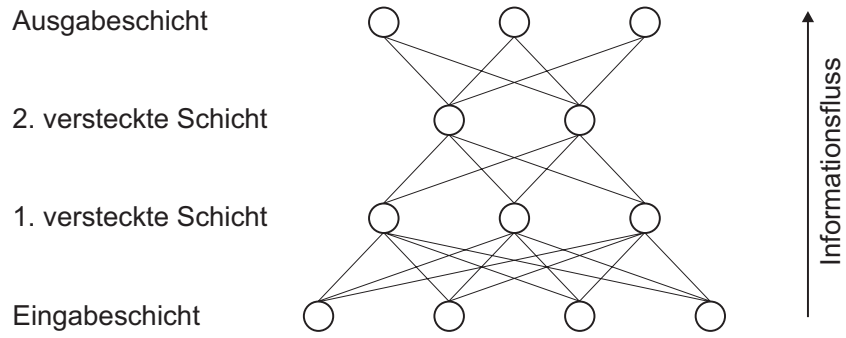


Abbildung 2.9: Beispiel für ein feedforward-Netz mit zwei versteckten Schichten und vollständiger Vernetzung

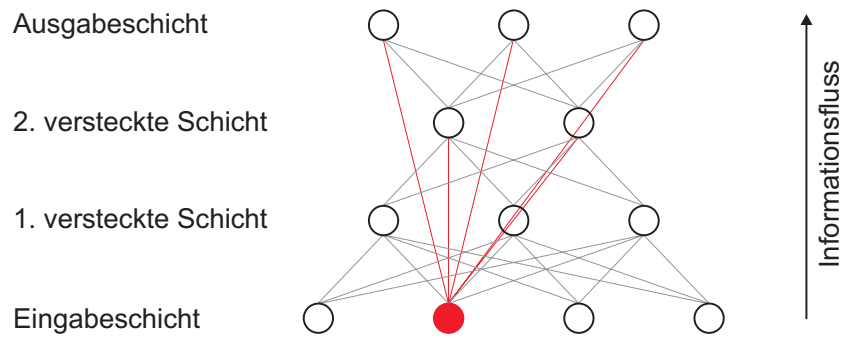


Abbildung 2.10: Exemplarische Darstellung einer Shortcut-Vernetzung eines Neurons

Da sie lediglich zur Informationsverarbeitung innerhalb des neuronalen Netzes dienen und deshalb für einen außenstehenden Betrachter nicht zu sehen sind, werden sie als *versteckte Neuronen* bezeichnet (Kinnebrock, 1994; Zell, 1994).

In Abbildung 2.9 handelt es sich um ein *vorwärtsgerichtetes Netz (feedforward-Netz)*, da der Informationsfluss lediglich in einer Richtung, und zwar von der Eingabe zur Ausgabe, stattfindet, d.h. jedem Eingabevektor wird ein Ausgabevektor zugeordnet. Außerdem ist dieses Netz *ebenenweise vollständig vernetzt*, d.h. alle Neuronen einer Schicht sind mit sämtlichen Neuronen der nächsthöheren Schicht verbunden. Neben der vollständigen Vernetzung gibt es noch die Möglichkeit der *Shortcut-Verbindungen*, bei denen zusätzlich Ebenen übersprungen werden können. Abbildung 2.10 zeigt ein solches Verbindungsschema. Hier sind die zusätzlichen Shortcut-Verbindungen rot dargestellt.

In sogenannten *rückgekoppelten Netzen (feedback-Netz)* sind zudem Verbindungen erlaubt, die die Ausgabe eines Neurons entweder auf dessen eigene Eingabe oder eines Neurons derselben oder einer niedrigeren Schicht³ lenken (vgl. Abbildung 2.11). Beim Anlegen einer Eingangsinformation an das Netz entsteht ein Iterationsprozess, der erst dann endet, wenn ein Gleichgewichtszustand erreicht ist (Zell, 1994).

³bei rückgekoppelten Netzen ist es oft nicht mehr möglich, von Schichten zu sprechen.

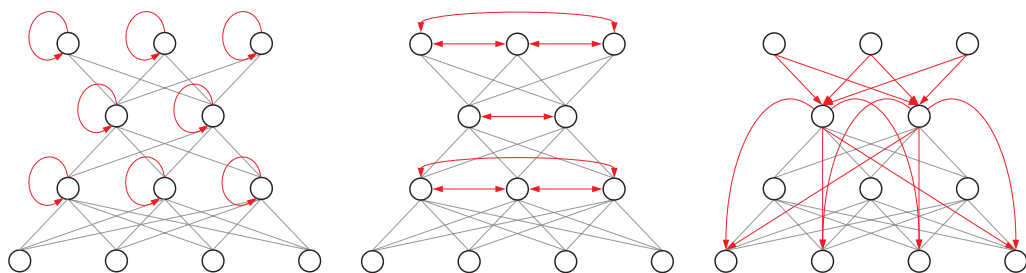


Abbildung 2.11: Beispiele für feedback-Verbindungen

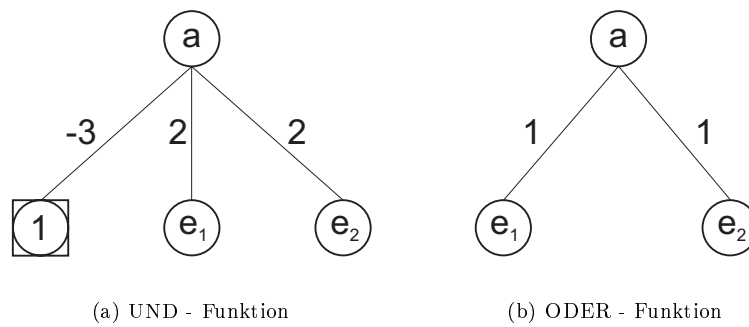


Abbildung 2.12: Beispiel: Bool'sche Funktionen

Ein kleines Beispiel zum besseren Verständnis

Die Bool'schen Funktionen *Und* und *Oder* können durch einfache neuronale Netze realisiert werden (siehe Abbildung 2.12). Dabei entsprechen die e_j den Eingabewerten und a dem Ausgabewert des Netzes. Das Quadrat symbolisiert ein zusätzliches Schwellwertneuron, die Werte an den Verbindungslinien die entsprechenden Gewichte. Als Aktivierungsfunktion verwenden wir die binäre Funktion. Das linke Verbindungsgewicht in Abbildung 2.12(a) stellt also einen Schwellwert dar. Da bei der *Oder*-Funktion kein Schwellwertneuron vorhanden ist, liegt der Schwellwert hier bei 0. Legen wir nun verschiedene Eingangswerte an, so entsprechen die Ausgabewerte den Ergebnissen der jeweiligen Bool'schen Funktion. *True* und *False* entsprechen den Werten 1 und 0. Eine Übersicht über alle möglichen Werte geben folgende Tabellen:

- Bool'sche Und-Funktion (vgl. Abbildung 2.12(a))

o_{e_1}	o_{e_2}	θ_a	net_a	o_a
0	0	1	$net_a = 0 \cdot 2 + 0 \cdot 2 + 1 \cdot (-3) = -3$	$o_a = f_{act}(-3) = 0$
0	1	1	$net_a = 0 \cdot 2 + 1 \cdot 2 + 1 \cdot (-3) = -1$	$o_a = f_{act}(-1) = 0$
1	0	1	$net_a = 1 \cdot 2 + 0 \cdot 2 + 1 \cdot (-3) = -1$	$o_a = f_{act}(-1) = 0$
1	1	1	$net_a = 1 \cdot 2 + 1 \cdot 2 + 1 \cdot (-3) = +1$	$o_a = f_{act}(+1) = 1$

- Bool'sche Oder-Funktion (vgl. Abbildung 2.12(b))

o_{e_1}	o_{e_2}	net_a	o_a
0	0	$net_a = 0 \cdot 1 + 0 \cdot 1 = 0$	$o_a = f_{act}(0) = 0$
0	1	$net_a = 0 \cdot 1 + 1 \cdot 1 = 1$	$o_a = f_{act}(1) = 1$
1	0	$net_a = 1 \cdot 1 + 0 \cdot 1 = 1$	$o_a = f_{act}(1) = 1$
1	1	$net_a = 1 \cdot 1 + 1 \cdot 1 = 2$	$o_a = f_{act}(2) = 1$

Wissen wird also in der Struktur der Netztopologie und in den Verbindungsgewichten gespeichert. Folgende Fragen stellen sich nun:

1. Welche Topologie wählen wir, und
2. wie finden wir die richtigen Gewichtswerte,

damit das Netz die gewünschte Information möglichst gut repräsentiert?

Wahl der Topologie

Für die erste Frage gibt es leider keine allgemein gültige Antwort. Natürlich ist die Anzahl der Eingabe- und Ausgabeneuronen durch die Aufgabenstellung festgelegt. Für die Zahl der versteckten Schichten, der darin jeweils enthaltenen versteckten Neuronen und die Art der Vernetzung gibt es dagegen keine strenge Festlegung, an die wir uns halten könnten oder müssten. Dies bedeutet, dass wir durch eine Vielzahl von Versuchen für jedes Problem eine eigene optimale Netzwerktopologie finden müssen. Glücklicherweise gibt es Anhaltspunkte, an denen wir uns orientieren können:

Cybenko (1989) hat gezeigt, dass jede kontinuierliche Funktion durch ein neuronales Netz mit nur *einer* versteckten Schicht beliebig genau repräsentiert werden kann. Genau *zwei* versteckte Schichten sind nötig, wenn die Funktion eine endliche Anzahl an Unstetigkeitsstellen besitzt (Conway, 1998). Die erzielbare Genauigkeit ist dabei lediglich von der Anzahl der versteckten Neuronen abhängig. Mit Hilfe dieser Aussage können wir die Auswahl möglicher Netzwerktopologien stark einschränken, da nur noch Netzwerke mit höchstens zwei versteckten Schichten untersucht werden müssen.

Trotzdem ist es schwierig, einen optimalen Netzaufbau zu finden. Zu wenig Neuronen oder Verbindungen führen zu einem eingeschränkten Bewegungsspielraum. Komplexe Systeme können somit vom Netz nicht mehr repräsentiert werden. Bei sehr vielen Verbindungen und damit sehr vielen zu bestimmenden Gewichten steigt dagegen der Rechenaufwand stark an. Es gilt also, einen Kompromiss zwischen diesen beiden Extrema zu finden. Auch für diese Problematik gibt es Orientierungshilfen. Eine Vorschrift für die Anzahl der zu verwendenden Neuronen ist in Otto (1995) zu finden. Dort wird eine Obergrenze von

$$h = \frac{\text{Anzahl der Trainingsdaten}}{5(n+m)} \quad (2.15)$$

mit n : Anzahl der Eingabeneuronen
 m : Anzahl der Ausgabeneuronen

angegeben (vgl. Heine, 1999). Auch diese Formel kann allerdings nur ein grober Anhaltspunkt sein und ersetzt keineswegs ein experimentelles Vorgehen beim Finden der optimalen Netztopologie.

Trainieren des Netzes

Die zweite Frage enthält die Forderung, die Gewichtswerte für jede einzelne Verbindung des gesamten Netzes so zu wählen, dass die vorliegende Aufgabenstellung gelöst wird, d.h. dass auf jede Netzeingabe die richtige Netzausgabe erfolgt. Das Anpassen der Gewichte wird *Training* oder *Lernen* genannt.

Um ein neuronales Netz zu trainieren, benötigen wir einen Trainingsdatensatz. Dieser enthält zu jedem Eingabedatensatz das richtige Ausgabemuster. Er sollte genug Muster enthalten, um alle Gewichte des Netzes bestimmen zu können (vgl. Unterbestimmung eines linearen Gleichungssystems). Stellen wir uns vor, unser Netz soll zwei Zahlen miteinander multiplizieren. Dann besteht ein Muster aus einem Zahlenpaar (Eingabe) und dessen Produkt (gewünschte Ausgabe). Der Trainingsdatensatz enthält eine Vielzahl dieser Muster.

Im ersten Schritt werden alle Gewichte des Netzes mit einem zufälligen kleinen Wert initialisiert, um einen neutralen Startpunkt für das Training zu erhalten. Als nächstes wird ein Muster aus dem Trainingsdatensatz ausgewählt und dessen Eingabewerte - in unserem Fall das Zahlenpaar - an den Eingabeneuronen angelegt. Mit den Formeln 2.8, 2.9 und 2.10 kann nun die Netzausgabe errechnet werden. Die Differenz zwischen dieser *Ist*-Ausgabe und der im Muster enthaltenen *Soll*-Ausgabe (Produkt) wird anschließend dazu verwendet, die Gewichte nach einem bestimmten Schema anzupassen. Eine *Trainingsiteration* ist dann vollendet, wenn dieser Vorgang für alle Muster des Trainingsdatensatzes einmal durchlaufen wurde. Um ein neuronales Netz zu trainieren, sind üblicherweise mehrere hundert oder sogar tausend Iterationen nötig.

Die mit Abstand populärste Methode, die Gewichte eines *feedforward*-Netzes mit dem Ziel der Fehlerminimierung anzupassen, ist das *Backpropagation*-Verfahren. Da es in dieser Arbeit überwiegend eingesetzt wird, soll es im folgenden kurz beschrieben werden:

Ausgangspunkt und Grundlage für die meisten komplizierteren Lernregeln ist die sogenannte *Hebbsche Lernregel*:

“Wenn Zelle j eine Eingabe von Zelle i erhält und beide gleichzeitig stark aktiviert sind, dann erhöhe das Gewicht w_{ij} (die Stärke der Verbindung von i nach j).” (Zell, 1994)

Die mathematische Darstellung lautet wie folgt:

$$\Delta w_{ij} = \eta o_i a_j \quad (2.16)$$

Dabei ist

- Δw_{ij} die Änderung des Verbindungsgewichtes w_{ij} ,
- η die sogenannte *Lernrate* (konstant),
- o_i die Ausgabe der Vorgängerzelle i und
- a_j die Aktivierung der Nachfolgerzelle j .

An dieser Stelle ist die Anlehnung der Mathematik an die Biologie wieder deutlich zu erkennen. Das Lernen beim Menschen erfolgt prinzipiell nach dem gleichen Schema (vgl. Kapitel 2.2.1). Für diese Formel gibt es noch eine allgemeinere mathematische Form:

$$\Delta w_{ij} = \eta h(o_i, w_{ij}) g(a_j, t_j) \quad (2.17)$$

Die Gewichtsänderung ist also zusätzlich Funktion des Verbindungsgewichtes w_{ij} zwischen Vorgängerzelle i und Zelle j und der erwarteten Aktivierung t_j (*teaching input*). Die erwartete Aktivierung kann bei den Ausgabeneuronen direkt aus dem Trainingsdatensatz ausgelesen werden.

Backpropagation ist eine Spezialisierung dieser allgemeinen Hebbischen Lernregel. Sie ermöglicht das Trainieren von Netzen mit mehreren versteckten Schichten:

$$\Delta w_{ij} = \eta o_i \delta_j \quad (2.18)$$

$$\delta_j = \begin{cases} f'_{act}(net_j) (t_j - o_j) & \text{falls } j \text{ ein Ausgabeneuron ist} \\ f'_{act}(net_j) \sum_k (\delta_k w_{jk}) & \text{falls } j \text{ ein verstecktes Neuron ist} \end{cases} \quad (2.19)$$

Die Gewichtsänderung ist proportional zum Ausgabefehler, der aus dem *teaching input* berechnet werden kann. Da nur für die Ausgabezellen ein *teaching input* aus dem Trainingsdatensatz vorliegt, muss unterschieden werden, ob es sich bei der Zelle j um eine Ausgabezelle oder eine versteckte Zelle handelt. Da für die Verbindungen, die nicht direkt in die Ausgabeschicht münden, kein direkter Vergleichswert aus dem Trainingsdatensatz und damit auch kein Fehler vorliegt, muss dieser aus allen Fehlern der direkten Nachfolgezellen berechnet werden. Dies ist in der Formel 2.19 durch die Summation über k ausgedrückt. Im ersten Schritt muss also die Gewichtsveränderung von der letzten versteckten Schicht zur Ausgabeschicht berechnet werden. Somit können wir uns für alle Zellen in der letzten versteckten Schicht Soll-Aktivierungen errechnen, die wir dann wiederum als *teaching input* für die vorletzte versteckte Schicht heranziehen. Dieser Vorgang wird solange wiederholt, bis wir in der Eingabeschicht angekommen sind und dort die Verbindungen zur ersten versteckten Schicht angepasst haben. Der Fehler wird somit, ausgehend von der Ausgabeschicht, für alle Zellen rekursiv berechnet. Dieses Zurückverfolgen des Fehlers gibt der Lernregel den Namen *Backpropagation-Regel* (Hoffmann, 1993, Kinnebrock, 1994; Zell, 1994).

Meist wird für f_{act} die logistische Aktivierungsfunktion (siehe Abbildung 2.8) verwendet

$$f_{act}(net) = \frac{1}{1 + e^{-c \cdot net}} \quad (2.20)$$

Deren Ableitung ergibt sich zu:

$$f'_{act}(net_j) = f_{act}(net_j) (1 - f_{act}(net_j)) = o_j (1 - o_j) \quad (2.21)$$

Somit erhalten wir durch Einsetzen von 2.21 in 2.19 schließlich für δ_j :

$$\delta_j = \left\{ \begin{array}{ll} c \cdot o_j (1 - o_j) (t_j - o_j) & \text{falls } j \text{ ein Ausgabeneuron ist} \\ c \cdot o_j (1 - o_j) \sum_k (\delta_k w_{jk}) & \text{falls } j \text{ ein verstecktes Neuron ist} \end{array} \right\} \quad (2.22)$$

Eine detaillierte Herleitung der Formel für die Backpropagation-Regel lässt sich in Zell (1994) finden. Lediglich auf die Verwendung der Ableitung soll kurz eingegangen werden: Backpropagation ist ein Gradientenabstiegsverfahren. Ziel dieses Verfahrens ist es, mit der Methode des steilsten Abstiegs möglichst rasch ein globales Minimum der

Fehlerfunktion zu finden. Die Fehlerfunktion können wir uns bei n zu bestimmenden Gewichten als n -dimensionale Hyperfläche vorstellen. Da wir bei mehr als 3 Dimensionen schnell an die Grenzen unserer Vorstellungskraft stoßen, soll unser kleines Beispiel (siehe Abbildung 2.12) das Prinzip verdeutlichen: Um die Bool'sche Oder-Funktion mit Hilfe eines neuronalen Netzes darstellen zu können, müssen genau zwei Gewichte bestimmt werden. Anfänglich sind diese mit zufälligen Werten belegt. Legen wir nun alle Muster unseren Trainingsdatensatzes am Netzeingang an, so erhalten wir nach Summieren und Quadrieren der Ausgabefehler den Gesamtfehler E des Netzes als skalaren Wert. Verändern wir die Gewichte, so wird sich auch der Netzfehler verändern. Er kann damit als Funktion zweier Variablen beschrieben werden:

$$E = f(w_1, w_2) \quad (2.23)$$

Um von einem beliebigen Punkt auf dieser Fehlerfläche zu einem Minimum zu gelangen, errechnen wir uns dort den Gradienten und bestimmen somit die Richtung des steilsten Abstiegs der Fehlerfunktion. Bildlich gesprochen gibt uns also die Ableitung $f'_{act}(net)$ die Richtung und der Faktor $(t_j - o_j)$ bzw. $\sum_k (\delta_k w_{ik})$ multipliziert mit dem Betrag des Gradienten die Länge unseres Weges an, den wir auf unserer Fehlerfunktion gehen müssen, um ein Minimum zu erreichen. Nicht immer handelt es sich dabei jedoch um das erwünschte globale Minimum. Da die Fehlerfläche meist sehr viele lokale Minima besitzt, haben alle Gradientenverfahren das Problem, dass sie in einem solchen hängenbleiben können. Praktische Erfahrungen haben allerdings gezeigt, dass Backpropagation bei sehr vielen Anwendungen ein Minimum findet, dessen Qualität nahe genug am globalen Minimum liegt (Zell, 1994).

Der Vollständigkeit halber sei noch erwähnt, dass es theoretisch noch weitere Arten gibt, ein neuronales Netz zu trainieren. So können wir uns vorstellen, dass neben den Gewichten auch die Aktivierungs- oder die Ausgabefunktion während des Lernens modifiziert wird. Ebenso denkbar ist das Entwickeln neuer Zellen oder das Löschen von Zellen innerhalb des Lernprozesses. Das letzte Verfahren ermöglicht es, neben der Einstellung der Gewichte gleichzeitig eine optimale Topologie für die gegebene Aufgabenstellungen zu finden.

2.2.3 Vor- und Nachteile

Neuronale Netze haben gegenüber herkömmlichen Ansätzen neben vielen positiven Eigenschaften auch gewisse Nachteile. Für ihre Anwendung ist es unerlässlich, ihre Leistungsfähigkeit richtig einschätzen zu können. Ein Ausschnitt der wichtigsten in Zell (1994) aufgeführten Kriterien zeigt eine Gegenüberstellung der positiven und negativen Eigenschaften und soll als Entscheidungshilfe dienen:

Vorteile neuronaler Netze

- *Lernfähigkeit:* Da sich neuronale Netze allein durch ein Lernverfahren an ihre jeweilige Aufgabe anpassen können, müssen die Algorithmen für unterschiedliche Aufgabenstellungen nicht umprogrammiert werden.
- *Parallelität:* Durch die massive Parallelität neuronaler Netze sind sie für die Implementierung auf Parallelrechnern sehr geeignet.
- *Hohe Fehlertoleranz:* Neuronale Netze haben den Vorteil, dass Information in den Gewichten verteilt gespeichert wird. Dadurch erreichen sie eine höhere Fehlertoleranz beim Ausfall einzelner Neuronen oder Verbindungen.
- *Höhere Robustheit gegenüber Störungen oder verrauschten Daten*
- *Generalisierung:* Oft bilden neuronale Netze beim Training automatisch Prototypen der Eingabemuster. Durch diese Generalisierung können später während der Anwendung auch nicht vollständig vorhandene Muster richtig klassifiziert werden.

Nachteile neuronaler Netze

- *Wissen kann ausschließlich durch Lernen erworben werden:* Es ist sehr schwer, einem neuronalen Netz eine Wissensbasis mitzugeben. In Einzelfällen ist es möglich, die Gewichte durch einen externen Algorithmus vorzubestimmen, ansonsten erfolgt der Wissenserwerb prinzipiell durch Lernen.
- *Wissen implizit gespeichert:* Neuronale Netze können ihr eigenes Wissen nicht analysieren. Auch für den Anwender ist es sehr schwer, die Art und Weise der Problemlösung nachzuvollziehen. Dies mag bei sehr kleinen Netzen mit wenigen Neuronen noch gelingen, größere Netze mit versteckten Schichten sind für den Anwender allerdings eine Black Box.

- *Relativ langsames Lernen:* Der zeitliche Lernaufwand steigt mit der Anzahl der Verbindungen stark an. Vor allem die vollständige Vernetzung mehrerer versteckter Schichten ist äußerst rechenintensiv.

Kapitel 3

Beschreibung von SNNS

3.1 Bestandteile des Software-Paketes *SNNS*

SNNS wurde am Institut für Parallele und Verteilte Höchstleistungsrechner der Universität Stuttgart entwickelt und dient der Simulation neuronaler Netze. Der Simulator besteht aus vier Hauptkomponenten: Simulator-Kern, Graphischer Benutzeroberfläche (GUI), BATCHMAN und einem Netzwerk-Compiler. Der Simulator-Kern greift direkt auf die Netz-Datenstrukturen zu und führt alle nötigen Manipulationen durch. Die GUI baut auf dem Simulator-Kern auf und steuert die Abläufe während der Simulation. Außerdem können wir mit Hilfe der GUI Netzwerke erstellen, verändern und visualisieren. Da das Lernen neuronaler Netze eine sehr rechenintensive Aufgabe ist, kann der Simulator auch über die Batch-Programmiersprache BATCHMAN gesteuert werden. Diese erlaubt die Abfrage und automatische Veränderung bestimmter Parameter während des Lernens ohne Benutzereingriff. Mit Hilfe des Netzwerk-Compilers ist es möglich, das neuronale Netz in Form von C-Quellcode darzustellen. Dieser kann dann in eigene Programme integriert werden. Im Folgenden werden nun die beiden für unsere Zielsetzung wichtigsten Bestandteile, die graphische Oberfläche und das Programm BATCHMAN beschrieben. Eine ausführliche Beschreibung ist in Zell et al. (1995) zu finden.

3.1.1 Graphische Oberfläche

Das Startfenster der GUI und die Bezeichnung der wichtigsten Schaltflächen ist in Abbildung 3.1 dargestellt.

Die Datenschnittstelle ermöglicht das Lesen und Schreiben von Dateien im Format *NET*, *PAT* (pattern) und *RES* (result):

- NET-Dateien enthalten die gesamte Netztopologie, die Aktivierungszustände der Neuronen, die Gewichte der Verbindungen und Parameter der Lernregel.
- PAT-Dateien (Pattern) beinhalten die Musterdatensätze zum Lernen bzw. Validieren.

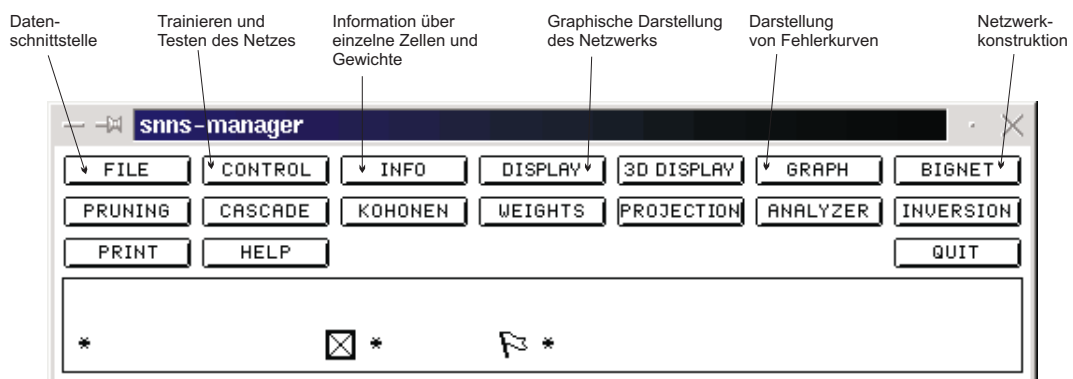


Abbildung 3.1: Das Startfenster: *snns-manager*

- RES-Dateien (Result) haben sowohl die Eingabewerte als auch die Soll- und Istausgabewerte des (trainierten) Netzes zum Inhalt. Diese Dateien können z.B. von anwendungsspezifischer Software ausgelesen werden und sind damit sehr wichtig für die spätere Analyse.

Zum Konstruieren des Netzwerks dient die Schaltfläche BIGNET. Hier können die verschiedenen Netzwerkkarten als Vorlagen gewählt und anschließend individuell angepasst werden. Der Aufbau der Netzwerktopologie wird exemplarisch anhand eines feedforward-Netzes gezeigt: Zunächst müssen die Anzahl der Schichten und der darin enthaltenen Neuronen festgelegt werden. Dann kann die Art der Verknüpfung gewählt werden. Neben den Vorgaben *FULL CONNECTION* (vollständige Vernetzung) und *SHORTCUT CONNECTION* (shortcut Verbindungen), die sich auf das gesamte Netz beziehen, können hier auch einzelne Verbindungen individuell erstellt oder gekappt werden. Neben den feedforward-Netzen werden noch weitere Netzwerktopologien auf ihre Verwendbarkeit zur Prädiktion der Erdrotationsparameter untersucht:

Netzwerk	Beschreibung
feedforward	(siehe Kapitel 2.2.2)
Jordan	Partiell rekurrentes Netzwerk. Erweiterung der feedforward-Netze um Kontextzellen, die zur Speicherung der Netzausgabe dienen. Die Netzeingabe setzt sich zusammen aus den Eingabeneuronen und den Ausgabewerten der Kontextzellen. Durch diese Rückkopplung erhält das Netz eine Art Gedächtnis, das bei der Prognose von Zeitreihen Vorteile bringen kann.
Elman	Partiell rekurrentes Netzwerk. Das Prinzip entspricht dem der Jordan-Netze. Der Unterschied besteht darin, dass nicht die Ausgabeschicht, sondern die versteckten Zellen mit den Kontextzellen verbunden sind. Die Ausgabe der Kontextzellen wird wiederum der Eingabe der selben versteckten Schicht zugeführt.

Das Trainieren und Testen des Netzes geschieht mit Hilfe des CONTROL-Fensters. Hier können z.B. die Trainingsmuster bzw. Validierungsmuster, die Anzahl der Trainingszyklen und die Reihenfolge der Musterpräsentation gewählt werden. Nur die Trainingsmuster werden zum Lernen der Verbindungsgewichte genutzt. Der Validierungsdatensatz enthält Muster, die im Trainingsdatensatz nicht vorkommen. Der errechnete Netzausgabefehler der Validierungsdaten dient der Formulierung eines Abbruchkriteriums für den Lernvorgang. Dadurch können wir verhindern, dass das Netz das in den Daten vorhandene Rauschen "auswendig" lernt und somit seine Generalisierungsfähigkeit verliert (Conway, 1998). Das Training wird also an der Stelle des kleinsten Validierungsfehlers abgebrochen.

Die Schaltfläche ALL startet das Training für die in dem Eingabefeld hinter CYCLES angegebene Anzahl an Trainingsiterationen. Dabei werden die Trainingsmuster dem Netz entweder in der vorliegenden Reihenfolge oder, wenn die Schaltfläche SHUFFLE aktiviert ist, zufällig präsentiert.

Im CONTROL-Fenster besteht außerdem die Möglichkeit, zwischen verschiedenen Lern-, Update- und Initialisierungsfunktionen zu wählen:

Durch die Wahl der Lernfunktion wird festgelegt, auf welche Art und Weise sich der Netzausgabefehler auf die Änderung der Verbindungsgewichte auswirken soll. Eine Auswahl der in dieser Arbeit verwendeten Funktionen zeigt folgende Tabelle:

Lernfunktion	Beschreibung
Std_Backpropagation	Standard-Backpropagation (siehe Kapitel 2.2.2)
BackpropMomentum	Backpropagation mit <i>Momentum-Term</i> und <i>Flat-Spot Elimination</i> reduziert die Wahrscheinlichkeit eines "Hängenbleibens" in einem lokalen Minimum.
BackpropBatch	Backpropagation, wobei die Gewichtsanzpassung erst nach dem Durchlauf aller Muster des Trainingsdatensatzes erfolgt
BackpropWeightDecay	Backpropagation mit Eliminierung "schwacher" Verbindungen
Quickprop	Mit Hilfe der zweiten Ableitung wird versucht, das globale Minimum der Fehlerfunktion schneller zu finden

Die Updatefunktionen bestimmen, in welcher Reihenfolge die Gewichtsanzpassung während des Lernens erfolgt. Die

Wahl der Updatefunktion hängt von der Topologie des Netzes ab, d.h. jeder Netzwerktyp benötigt eine spezielle Art der Gewichts Anpassung:

Updatefunktion	Netztyp	Beschreibung
JE_Order	Jordan und Elman	Das Eingabemuster wird bis zur Ausgangsschicht propagiert. Die Kontextzellen erhalten anschließend gleichzeitig ihre neuen Werte.
JE_Special	Jordan und Elman	Das Eingabemuster wird dynamisch aus den letzten Ausgabewerten der Ein- und Ausgabezellen erstellt.
Synchronous_Order	beliebig	Alle Neuronen erhalten gleichzeitig neue Werte
Topological_Order	beliebig	Die neue Aktivierung wird in topologischer Reihenfolge errechnet. Zuerst werden die Neuronen der Eingabeschicht aktualisiert, anschließend die der 1. versteckten Schicht und zum Schluss die Ausgabeneuronen.

Die Verbindungsgewichte müssen vor dem ersten Lernzyklus mit Startwerten belegt werden. Es genügt allerdings nicht, alle Gewichte mit dem gleichen Zahlenwert zu versehen. Um sinnvoll zu arbeiten, braucht z.B. Backpropagation uneinheitliche Gewichtswerte. Für die Initialisierung der Verbindungsgewichte vor dem Trainieren sorgt die Initialisierungsfunktion.

Initialisierungsfunktion	Netztyp	Beschreibung
Randomize_Weights	beliebig	Initialisiert alle Gewichte (inklusive Schwellwert) mit zufälligen Zahlen aus einem bestimmten Intervall
JE_Weights	Jordan und Elman	Unterscheidung zwischen herkömmlichen Zellen und Kontextzellen

Über die Schaltfläche INFO im MANAGER-Fenster können Informationen über den Aktivierungszustand einzelner Zellen und sämtliche Verbindungsgewichte abgefragt und individual verändert werden. Außerdem ist es möglich, die Aktivierungs- und Ausgabefunktion für jede Zelle festzulegen. Voreingestellt sind für die Aktivierungsfunktion die logistische Funktion und für die Ausgabefunktion die Identität.

Der GRAPH-Button ermöglicht eine visuelle Kontrolle des Lernfortschritts, indem der Lern- und/oder Validierungsfehler in Abhängigkeit der Zeit aufgetragen werden. Diese Funktion hat allerdings in der zur Verfügung stehenden Version 4.1 noch einige Fehler, so dass über die BATCHMAN-Schnittstelle ein eigenes Programm zur Darstellung der Fehlerkurven erstellt werden musste.

3.1.2 BATCHMAN

Diese Batch-Programmiersprache unterstützt alle wichtigen Funktionen, die zum Trainieren und Testen eines neuronalen Netzes notwendig sind. So sind alle nicht-graphischen Funktionen der GUI verfügbar. Die Terminologie der Sprache ist vergleichbar mit der von Hochsprachen wie BASIC, PASCAL oder C. Die wichtigsten Programmierbausteine, wie Schleifen, logische Abfragen und Funktionen zur Ein- und Ausgabe sind vorhanden. Außerdem ist ein Aufruf von UNIX-Befehlen innerhalb eines Batch-Programmes möglich. Durch diese externe Kommunikationsfähigkeit ist es z.B. möglich, eigene C-Programme während des Lernvorgangs aufzurufen, um Analysefunktionen auszuführen oder graphische Ausgaben des Lernfortschritts zu erstellen.

3.2 Beschreibung der Funktionsweise anhand einfacher Beispiele

Um ein Gefühl für die benötigte Netztopologie, das ökonomischste Lernverfahren und die effektivste Mustereinteilung zu erlangen, werden die Prädiktionseigenschaften zunächst mittels einfacher Funktionen untersucht.

3.2.1 Allgemeines zur Vorhersagemethode

Vorhersagestrategien

Drei unterschiedliche Vorhersagestrategien werden analysiert:

1. Die Zeitreihen können mathematisch beschrieben werden durch $EOP = f(t)$. Wir können uns also ein neuronales Netz vorstellen, das nur ein Eingabeneuron und ein Ausgabeneuron besitzt: Am Eingabeneuron wird die Zeitvariable t angelegt, das Ausgabeneuron soll dann den entsprechenden Funktionswert $f(t)$ als Ausgabewert liefern. Die zwei großen Vorteile dieser Methode sind zum einen die Fähigkeit, ohne Kenntnis der Funktionswerte $f(t_{1..n})$ den Wert zum Zeitpunkt t_{n+1} vorhersagen zu können und zum anderen die Möglichkeit, nicht gleichabständige Zeitreihen zu verwenden.
2. In einem sukzessiven Verfahren werden an ein Netz mit n Eingabeneuronen die Funktionswerte der letzten bekannten Zeitpunkte $f(t_0), f(t_{-1})$ bis $f(t_{-(n-1)})$ angelegt und an einem einzigen Ausgabeneuron der Funktionswert des nächsten Zeitpunktes $f(t_1)$ abgelesen. Letzterer dient anschließend zusammen mit den letzten $n - 1$ bekannten Werten wiederum als Eingabe für die Vorhersage von $f(t_2)$. Die nächsten Eingabewerte setzen sich dann zusammen aus den bekannten $n - 2$ Werten und den bereits vorhergesagten Werten $f(t_1)$ und $f(t_2)$ und resultieren in einer Vorhersage von $f(t_3)$. Diesen Vorgang können wir für jeden gewünschten Vorhersagezeitraum entsprechend oft wiederholen. Der Nachteil im Vergleich zur ersten Strategie ist, dass die Funktionswerte hier gleichabständig vorliegen müssen, da die Zeitinformation nicht direkt berücksichtigt werden kann.
3. Wiederum werden an ein Netz mit n Eingabeneuronen die Funktionswerte der letzten bekannten Zeitpunkte $f(t_0), f(t_{-1})$ bis $f(t_{-(n-1)})$ angelegt, aber an den jetzt insgesamt m Ausgabeneuronen die zukünftigen Funktionswerte $f(t_1), f(t_2)$ bis $f(t_m)$ direkt abgelesen. Auch hier ist es notwendig, dass die Funktionswerte gleichabständig vorliegen. Außerdem ist der Vorhersagezeitraum begrenzt, da der Funktionswert maximal für den Zeitpunkt t_m bestimmt werden kann.

Die Darstellung dieser *Mustereinteilungen*, wie sie in den Vorhersagestrategien 2 und 3 verwendet werden, geschieht folgendermaßen:

Eingabemuster: $[t_0, t_{-1} \dots t_{-(n-1)}]$

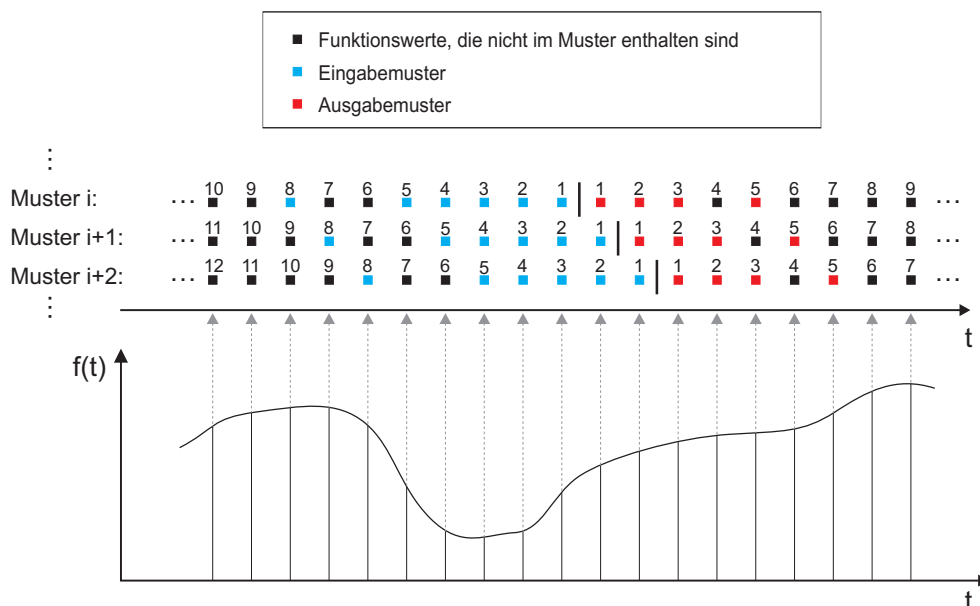
Ausgabemuster: $[t_1, t_2 \dots t_m]$

Wenn z.B. aus den Werten der letzten 5 Tage und des 8 Tage zurückliegenden Wertes auf die zukünftigen Werte in 1, 2, 3 und 5 Tagen geschlossen werden soll, so wird dies dargestellt durch:

Eingabemuster: $[1 \dots 5 \ 8]$

Ausgabemuster: $[1 \dots 3 \ 5]$

Folgende Zeichnung soll die Systematik verdeutlichen:



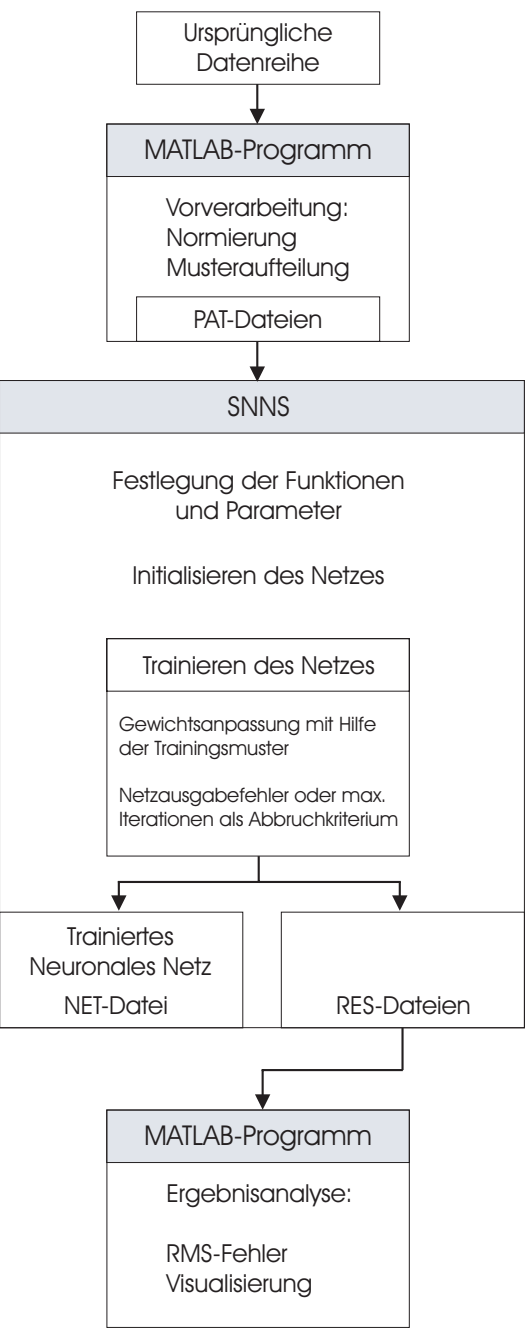


Abbildung 3.2: Datenfluss im Überblick

RMS-Fehler

Zur Analyse der Vorhersagequalität wird die statistische Größe *Root-Mean-Square* (RMS) herangezogen. Angenommen, wir hätten n Ausgabenneuronen, die Soll-Ausgabewerte seien t_j , die tatsächlichen Ist-Ausgabewerte o_j , mit $j = 1 \dots n$. Dann ist der RMS-Fehler

$$RMS = \sqrt{\frac{\sum_{j=1}^n (o_j - t_j)^2}{n}}. \tag{3.1}$$

Diese Formel erinnert stark an die Berechnung der Standardabweichung, bei der im Zähler anstatt eines Soll-Wertes t_j das arithmetische Mittel und im Nenner anstatt des n ein $n - 1$ steht. Letzteres wirkt sich bei großen Datemengen allerdings nicht aus.

Oft wird der RMS-Fehler nicht für jedes Ausgabemuster, sondern für jeden Vorhersagetag berechnet. Diese Berechnung erfolgt analog, wobei dann nicht über die Ausgabezellen, sondern über alle Muster summiert wird.

Mustersaufteilung und Datenfluss

Abbildung 3.2 zeigt vereinfacht den Datenfluss von der Datenreihe bis zur Ergebnisanalyse im Überblick.

Die zu präzifizierende Datenreihe wird zunächst mit Hilfe eines MATLAB-Programmes auf den Wertebereich $[0;2;0.8]$ normiert. Dies ist nötig, da die logistische Aktivierungsfunktion nur einen Wertebereich von $[0;1]$ und daher große Zahlen vom Netz nicht repräsentiert werden können. Außerdem ist dadurch trotz unterschiedlicher Daten ein direkter Fehlervergleich möglich. Das Aussparen der Randbereiche auf einer Breite von 0.2 hat den Vorteil, dass die zu approximierende Funktion vom Netz leichter repräsentiert werden kann.

Je nach verwendeter Vorhersagestrategie werden nun aus der normierten Datenreihe Muster erstellt. Jedes Muster enthält die Datenwerte für Ein- und Ausgabenneuronen. Durch diesen Schritt wird automatisch auch die Anzahl der Eingabe- und Ausgabenneuronen des neuronalen Netzes festgelegt. Anschließend wird die Gesamtmenge aller Muster Ω prozentual in drei Teilmengen Ω_{TRAIN} , Ω_{TEST} und Ω_{VALID} zerlegt (siehe Abbildung 3.3). Dabei wird zunächst festgelegt, bis zu welcher Epoche t_{train} die Daten zum Trainieren des Netzes verwendet werden sollen. Aus dieser Menge ($\Omega_{TRAIN} \cup \Omega_{TEST}$) wird dann zufällig ein (meist kleiner) Prozentsatz an Muster herausgegriffen und der Teilmenge Ω_{TEST} zugeschrieben. Alle verbleibenden Muster bilden die Teilmenge Ω_{TRAIN} . Diejenigen Muster, die zeitlich hinter der Epoche t_{train} liegen, bilden die Teilmenge Ω_{VALID} . Die *Mustersaufteilung* geschieht also so, dass die drei Teilmengen kein gemeinsames Muster besitzen und daher disjunkt sind.

Der Zweck dieser Aufteilung besteht - wie bereits in Kapitel 3.1.1 erwähnt - darin, eine Aussage über die Generalisierungsfähigkeit des Netzes machen zu können. Zum Trainieren werden ausschließlich Muster der Teilmenge Ω_{TRAIN} herangezogen. Die Testmuster, die beliebig zwischen die Trainingsmuster gestreut werden, dienen dazu,

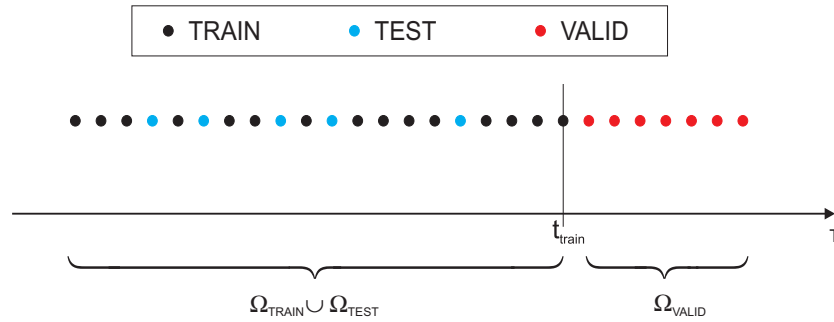


Abbildung 3.3: Musteraufteilung in Trainings-, Test- und Validierungsmuster

die Interpolationsfähigkeit des Netzes zu überprüfen. Mit Hilfe der Teilmenge Ω_{VALID} können wir dagegen auf die Extrapolationsfähigkeit des Netzes schließen und somit die Prädiktionsqualität bestimmen.

Die Epoche t_{train} entspricht im Anwendungsfall bei der Vorhersage dem Zeitpunkt, für den der letzte Funktionswert bekannt ist.

Im Anschluß an die Erstellung der Muster werden diese im PAT-Format gespeichert.

Es ist wichtig, zwischen *Musteraufteilung* und *Mustereinteilung* zu unterscheiden: Während die *Musteraufteilung* festlegt, wie die Gesamtmenge an generierbaren Mustern auf die drei Musterteilmengen aufgeteilt wird, bestimmt die *Mustereinteilung*, aus welchen Werten der Vergangenheit jeweils auf welche Werte der Zukunft geschlossen werden soll und determiniert dadurch die Anzahl an Ein- und Ausgabeneuronen des Netzes.

Nach der Konstruktion der gewünschten Netztopologie mit dem BIGNET-Tool können wir uns entscheiden, ob wir die GUI oder BATCHMAN für das Training nutzen wollen. Bevor das Training gestartet wird, müssen zunächst die drei Musterdatensätze eingelesen werden, die Lern-, Update- und Initialisierungsfunktionen gewählt und deren Parameter festgelegt werden. Außerdem müssen an dieser Stelle die Reihenfolge der Präsentation der Trainingsmuster und innerhalb der GUI die maximale Anzahl der Trainingszyklen definiert werden. Nach der Initialisierung der Gewichte kann mit dem Training begonnen werden.

Nach jeder Trainingsiteration, d.h. einmaligem Durchlaufen aller Trainingsmuster, werden die Netzausgabefehler der Test- und Validierungsmuster berechnet und ausgegeben. Das Training ist dann beendet, wenn die Netzausgabefehler von Test- und Validierungsmuster minimal sind. Diese Abfrage ist allerdings nur im batch-Modus möglich. Dadurch soll ein Übertrainieren des Netzes verhindert werden, bei dem der Trainingsfehler zwar weiter fallen, Test- und Validierungsfehler aber ansteigen würden. Ideal wären gleichgroße Fehler für alle drei Mustermengen. Innerhalb der GUI bestimmt nur die maximale Anzahl an Iterationszyklen das Ende des Trainings und verhindert damit eine flexible Steuerung des Lernvorgangs.

Das fertig trainierte Netz kann dann als NET-Datei gespeichert werden. Die Ausgaben, die das Netz beim Anlegen der Mustereingaben liefert, werden für jede Mustermenge separat in drei RES-Dateien gespeichert. Diese RES-Dateien werden wiederum in MATLAB eingelesen, visualisiert und analysiert.

Der Datenfluss beim sukzessiven Verfahren ist um einiges komplexer (siehe Abbildung 3.4). Deshalb wird die Funktionsweise der erstellten C-Software kurz vorgestellt. Bis zur Generierung der RES-Dateien entspricht der Ablauf dem oben beschriebenen Verfahren. Anschließend wird die C-Software gestartet. Diese extrahiert aus der erzeugten RES-Datei der TRAIN-, TEST- oder VALID-Muster das erste Muster und speichert es in einer iterativen PAT-Datei. Anschließend wird innerhalb des C-Programmes ein batch-Programm gestartet, das über BATCHMAN dieses Muster am Netzeingang anlegt und den Soll- und Ist- Ausgabewert des Netzes in der iterativen RES-Datei speichert. Aus dem Ist-Ausgabewert wird ein neues Muster erzeugt und wiederum in der iterativen PAT-Datei gespeichert, so dass diese immer nur ein Datenmuster enthält. Dies wird solange fortgesetzt, bis das Ende des Vorhersagezeitraums erreicht ist. Anschließend wird dann für jeden Vorhersagezeitpunkt die quadratische Differenz zwischen Soll- und Ist-Ausgabe berechnet.

Diesen Vorgang können wir für jedes Muster aus der originalen RES-Datei wiederholen, indem wir es als initiales Muster der sukzessiven Vorhersage benutzen. Dadurch ist es möglich, für jeden Vorhersagezeitpunkt einen RMS-Fehler zu berechnen.

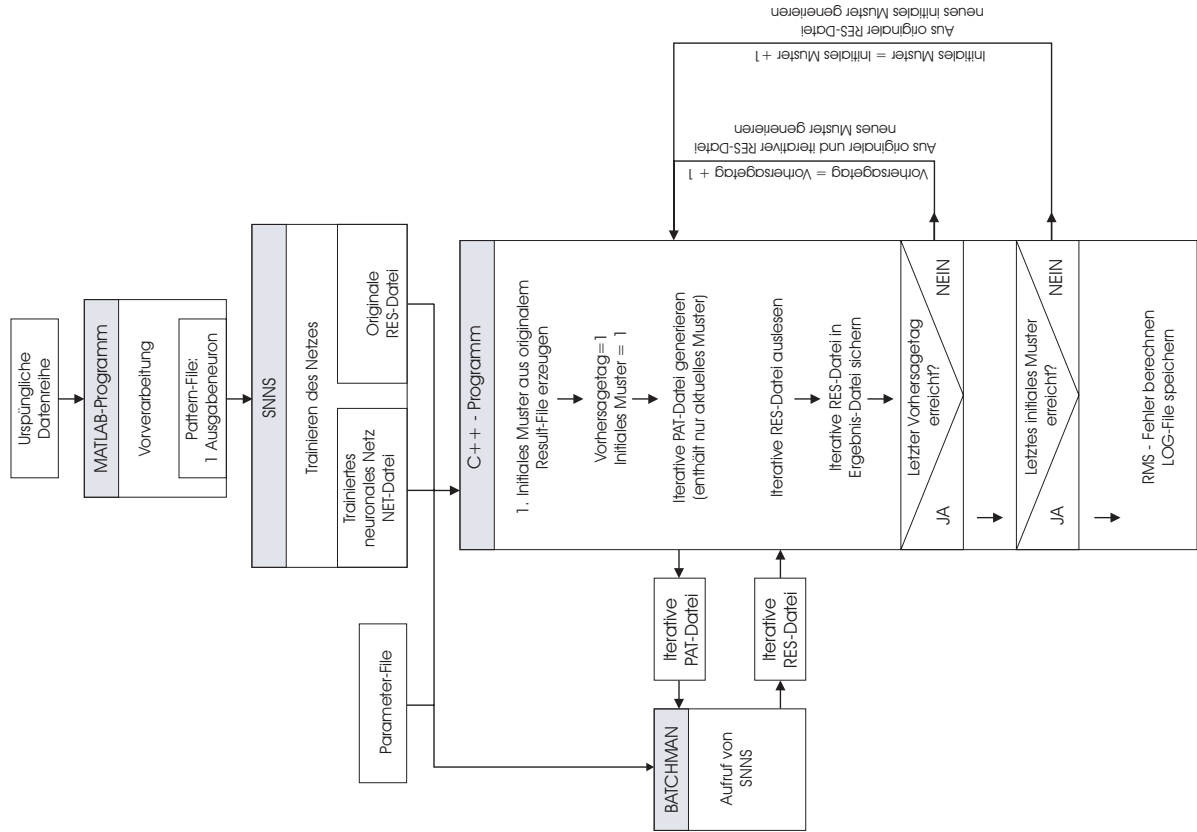


Abbildung 3.4: Schema der sukzessiven Vorhersage

3.2.2 Vorhersage einer Geraden

Anhand einer Geraden mit der Gleichung $y = t$ mit $t \in [0; 1]$ sollen die Prädiktionseigenschaften und die Auswirkungen verschiedener Konfigurationen getestet werden. Dazu tasten wir die Gerade im Definitionsbereich an 1000 äquidistanten Stützstellen ab.

Konfiguration 1¹

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	1	
Musteraufteilung	TRAIN+TEST: 95% davon TRAIN: 95% VALID: 5%	950 Muster 903 Muster 50 Muster
Eingabemuster	t	$n=1$
Ausgabemuster	y	$m=1$
Netzarchitektur	feedforward: 1-91-1	
Verknüpfung	vollständig	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.5$
Updatefunktion	Topological_Order	
Initialisierungsfunktion	Randomize_Weights	Intervall $[-1; 1]$
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	zufällig	

Da wir für die Vorhersagestrategie 1 nur die Zeit als Eingabe und den Funktionswert als Ausgabe benutzen, ist die

¹Zur Erläuterung der verschiedenen Einstellmöglichkeiten siehe Kapitel 2.2.2 und 3.2.1.

Anzahl der Ein- und Ausgabeneuronen bereits auf 1 festgelegt. Weil es sich um eine stetige Funktionen handelt, genügt es, eine versteckte Schicht zu verwenden. Nach Formel 2.15 benötigen wir maximal $h = \frac{903}{5(1+1)} \approx 91$ Neuronen. Die Netzarchitektur können wir deshalb in Kurzform schreiben als: 1-91-1. Die anfängliche Lernrate von 0.5 wurde nach 5000 Trainingsiterationen auf 0.3 und nach 7000 Iterationen auf 0.1 herabgesetzt, um ein Oszillieren des Netzausgabefehlers zu vermeiden. Den Netzausgabefehler während des Trainierens und die Approximation der Geraden durch das neuronale Netz nach 8000 Trainingsiterationen zeigt Abbildung 3.5. Zur Analyse des Trainingsfortschrittes wurden alle der ersten 50 Trainingsiterationen ausgewertet. Anschließend wurden die Fehler lediglich jeder 50. Iteration berechnet und gesichert. Sehr auffällig ist, dass der Fehler aller Musterteilmengen, wenn wir von einem untrainierten initialisierten Netz ausgehen, bereits nach den ersten 5 Iterationen auf ca. 10 Prozent fällt. Ein weiterer Fehlerrückgang bedarf bei fortschreitendem Training einer immer größeren Anzahl an Trainingsiterationen. Nach 8000 Iterationen ist der RMS-Fehler des Trainingsdatensatzes auf 2 Prozent gesunken. Der Fehlerverlauf der TEST-Muster ist unruhiger, aber entspricht betragsmäßig dem der TRAIN-Muster. Daraus können wir folgern, dass die Interpolationsfähigkeit des Netzes annähernd so gut ist, wie die Approximationsfähigkeit der trainierten Funktionswerte. Der Fehler der VALID-Muster, der die Extrapolations- bzw. Prädiktionsqualität beschreibt, ist dagegen vergleichsweise groß und zeigt einen weitaus unruhigeren Verlauf. Die RMS-Fehler nach 8000 Iterationen sind:

RMS_{TRAIN}	0.0086
RMS_{TEST}	0.0088
RMS_{VALID}	0.0273

Deutlich sind auch die Sprünge der Kurven bei 5000 und 7000 Trainingsiterationen erkennbar. Durch das Herabsetzen der Lernrate an diesen Stellen sind die Veränderungen der Gewichte pro Trainingsiteration nicht mehr so groß. Der dadurch hervorgerufene glattere Verlauf ist vor allem bei den TEST-Mustern feststellbar.

Der Vergleich der vom Netz gelieferten Daten mit der Referenzgeraden zeigt vor allem im oberen und unteren Wertebereich trotz Aussparung der Randzonen immer noch größere Abweichungen. Diese Abweichungen werden mit zunehmenden Trainingsiterationen zwar kleiner, bleiben aber aufgrund der Nichtlinearität der Übertragungsfunktion unvermeidbar.

Die Prädiktionsqualität nimmt innerhalb der ersten 50 Vorhersagezeitpunkte linear ab. Bei Vorhersagen, die sich über diesen Zeitraum hinaus erstrecken, wächst der Fehler dagegen sehr viel schneller. Diese Vorhersagestrategie scheint deshalb für die Prädiktion der langfristigen Entwicklung der Erdorientierungsparameter ungeeignet.

Konfiguration 2

Der einzige Unterschied zur Konfiguration 1 besteht darin, dass die Trainingsmuster während des Lernvorgangs nicht zufällig, sondern in chronologischer Reihenfolge präsentiert werden. Erstaunlicherweise führt das zu einem vollkommen anderen Trainingsverlauf. Die RMS-Fehler nach 8000 Iterationen sind hier:

RMS_{TRAIN}	0.0130
RMS_{TEST}	0.0132
RMS_{VALID}	0.0086

Wie Abbildung 3.6 erkennen lässt, ist der Lernerfolg am Anfang sehr gering. Erst nach 100 Trainingsiterationen ist ein wesentlicher Fehlerrückgang festzustellen. Dies liegt wahrscheinlich daran, dass die Unterschiede zweier aufeinanderfolgender Muster nur sehr gering sind und deshalb die Anpassung der Gewichte sehr langsam erfolgt. Daher zeigen die Fehlerkurven auch einen sehr glatten Verlauf.

Die Kurven der TRAIN- und TEST-Muster haben wie bei Konfiguration 1 einen ähnlichen Verlauf, haben aber etwas größere Werte. Sehr auffällig ist jedoch, dass der Fehler der VALID-Muster im Gegensatz dazu geringer ist. Dies bestätigt der Prädiktionsfehler in der untersten Graphik, der nur ca. 40 Prozent des Prädiktionsfehlers von Konfiguration 1 beträgt. Die Abweichung der vom Netz gelieferten Daten von der Referenzgeraden ist ebenfalls geringer. Außerdem ist diese Kurve etwas zugunsten der prädizierten Daten verschoben, womit der kleinere Prädiktionsfehler zu erklären ist.

Die Vorhersage ist also besser als die Approximation der Trainingsdaten selbst. Zu erklären ist dies eigentlich nur mit einer sehr günstigen Initialisierung. Trainieren wir das Netz weiter, so verschiebt sich das Gleichgewicht langsam zugunsten der TRAIN- und TEST-Muster. Die zunächst also als günstiger erscheinende Konfiguration erweist sich somit bei näherer Betrachtung als weniger vorteilhaft, da der Lernfortschritt im Vergleich zur Konfiguration 1 wesentlich langsamer verläuft.

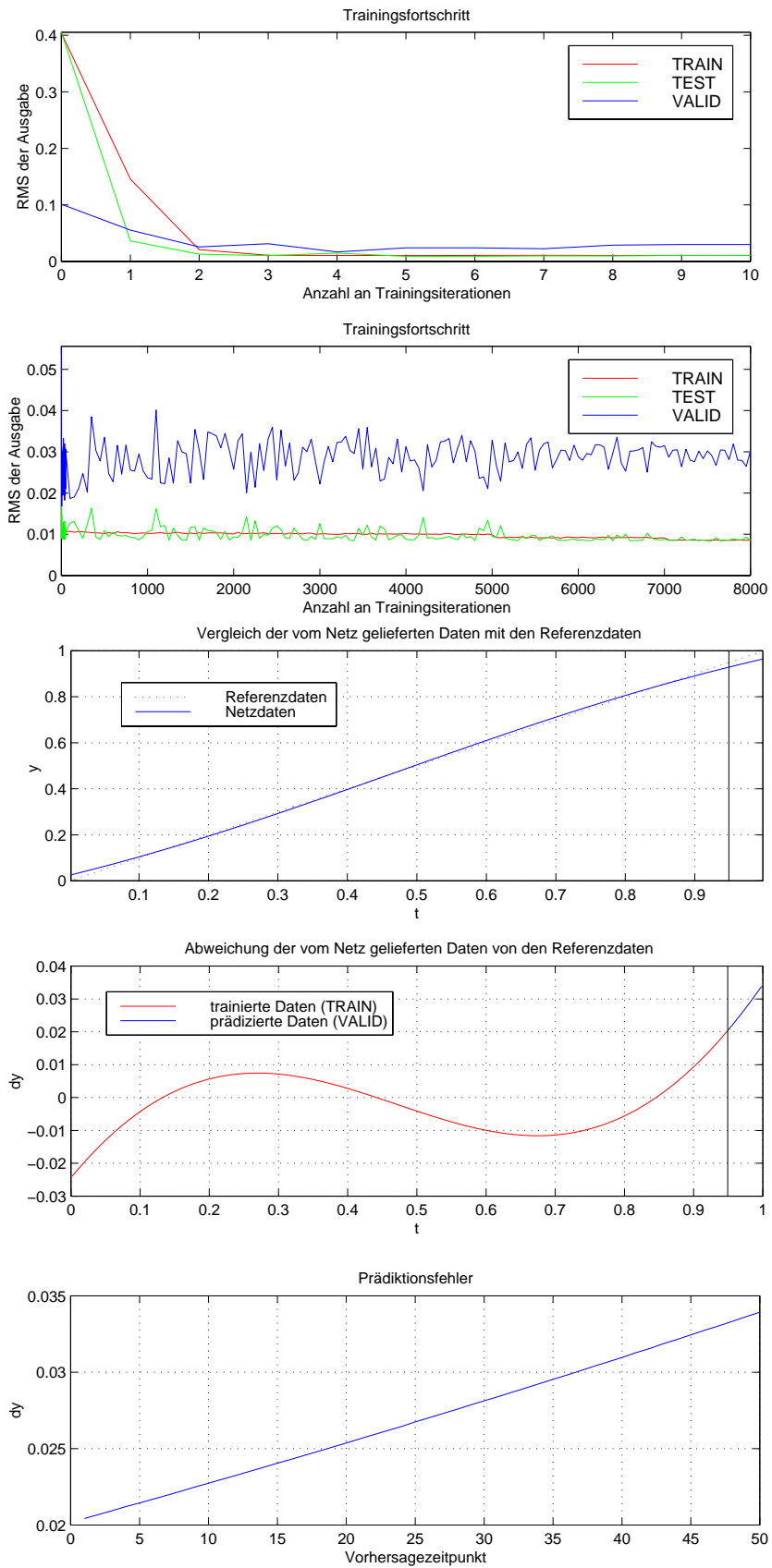


Abbildung 3.5: Trainingsfortschritt und Geradenapproximation Konfiguration 1

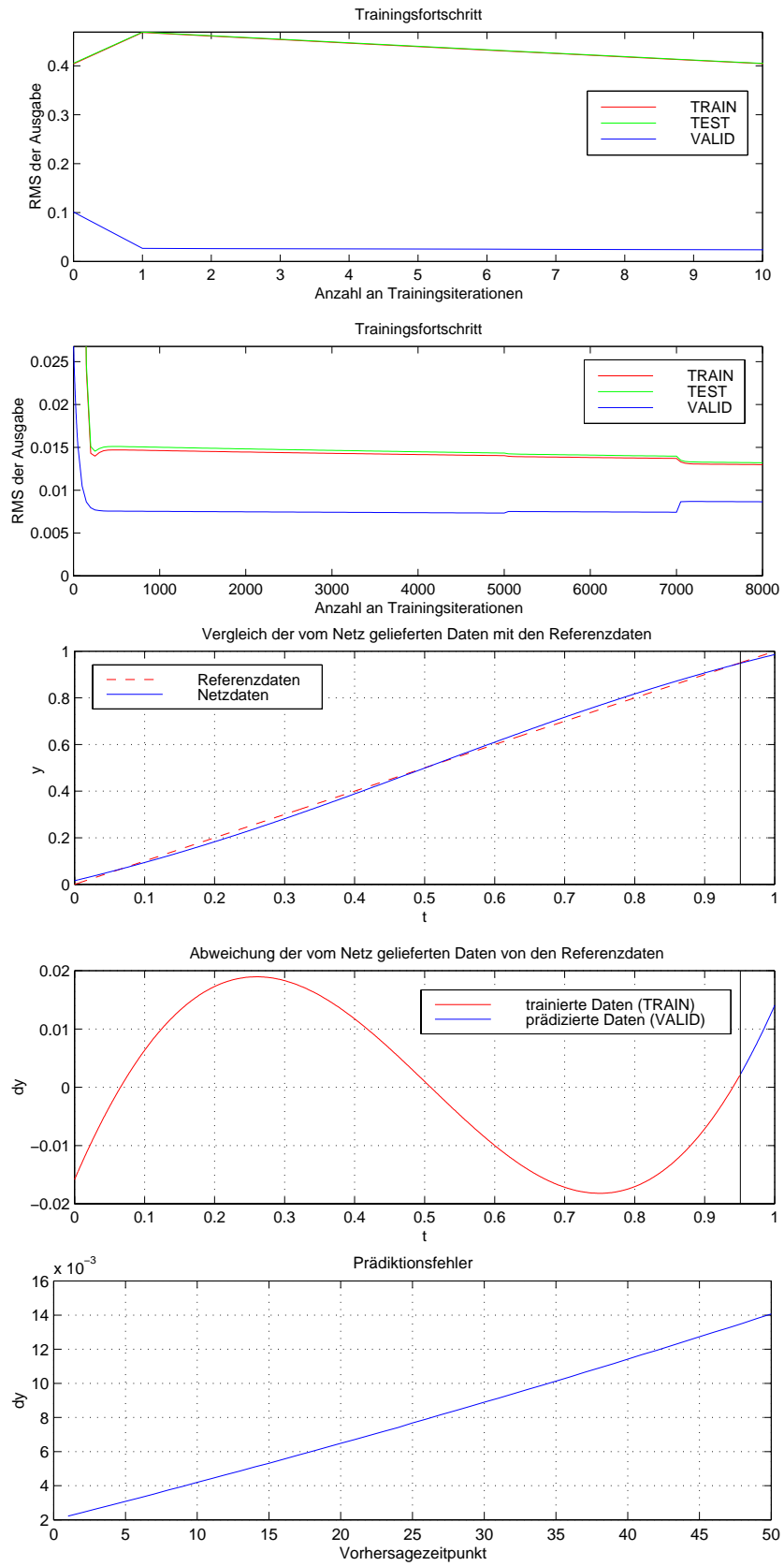


Abbildung 3.6: Trainingsfortschritt und Geradenapproximation Konfiguration 2

Konfiguration 3

Ausgehend von Konfiguration 1 überprüfen wir den Einfluss der Anzahl versteckter Neuronen mit Hilfe des RMS-Fehlers nach 8000 Iterationen:

versteckte Neuronen	90	45	20	10	5	4	3	1
RMS_{TRAIN}	0.0086	0.0087	0.0088	0.0049	0.0008	0.0007	0.0064	0.0097
RMS_{TEST}	0.0088	0.0088	0.0091	0.0050	0.0007	0.0007	0.0066	0.0099
RMS_{VALID}	0.0273	0.0276	0.0277	0.0183	0.0062	0.0057	0.0223	0.0328

Durch die richtige Anzahl an Neuronen kann der Lernaufwand offensichtlich entscheidend verringert werden. Bei nur vier versteckten Neuronen ist der RMS-Fehler minimal. Das Ergebnis ist in Abbildung 3.7 zu sehen. Die Gerade wird sehr schnell sehr gut approximiert. Der Vorhersagefehler steigt zwar relativ stark an, ist aber wesentlich geringer als bei Konfiguration 1.

Konfiguration 4

Die Auswirkung verschiedener Lernraten werden mit Konfiguration 3 als Grundlage getestet. Dazu wurde der Lernvorgang jeweils bei einem RMS-Fehler von 0.0007 abgebrochen und die Anzahl der Trainingsiterationen notiert:

Lernrate	0.3	0.5	0.7	0.9	1.0
Anzahl benötigter Trainingsiterationen	15100	7900	6000	4600	4800

Folglich kann auch durch die richtige Wahl der Lernrate der Trainingsaufwand wesentlich verkürzt werden.

Konfiguration 5

Hier sollen die Wirkungen unterschiedlicher Lern- und Updatefunktionen analysiert werden:

Verwenden wir anstatt der Lernfunktion *Topological_Order* die Funktion *Synchronous_Order*, so wirkt sich dies nicht merkbar auf das Ergebnis aus. Auch die Verwendung der in Kapitel 3.2.1 beschriebenen Lernfunktionen bringt gegenüber dem *Std_Backpropagation* keine Vorteile. Ihr Nutzen in diesem Beispiel ist fraglich, da zum einen weder die Genauigkeit noch die Lerngeschwindigkeit erhöht werden und zum anderen weitere funktionscharakteristische Parameter festgelegt werden müssen, die das Finden einer optimalen Konfiguration wiederum erschweren.

Die Lernfunktion *BackpropMomentum* erleichtert das schnelle Auffinden des globalen Minimums der Fehlerfläche. Dies hat zur Folge, dass der RMS-Fehler des Trainingsdatensatzes zügiger kleiner wird. Bei der Verwendung ist allerdings Vorsicht geboten, denn dadurch kann das neuronale Netz einen Teil seines Generalisierungsvermögens verlieren, da nun auch das "Rauschen" in den Daten schneller mitgelernt wird. Deshalb ist es in manchen Fällen nicht nur unnötig, sondern sogar für die Aufgabe der Prädiktion unvorteilhaft, das globale Minimum finden zu wollen.

Konfiguration 6

Die Auswirkungen der Vorhersagestrategie 3 sind Gegenstand dieser Konfiguration. Der Grund für die spätere Behandlung von Strategie 2 ist, dass diese auf Strategie 3 aufbaut und daher nur eine Erweiterung darstellt. Die Vorhersagestrategie 3 besteht darin, die nächsten Funktionswerte aufgrund der vorhergehenden zu bestimmen. Dazu verwenden wir als Eingabe die vergangenen drei Werte und als Ausgabe die drei folgenden. Die maximale Anzahl der zu bestimmenden Gewichte liegt damit bei $h = \frac{898}{5(3+3)} \approx 30$. Beim Trainingsstart wird eine Lernrate von $\eta = 0.9$ verwendet. Diese wird, wie oben beschrieben, während des Trainierens stufenweise abgesenkt.

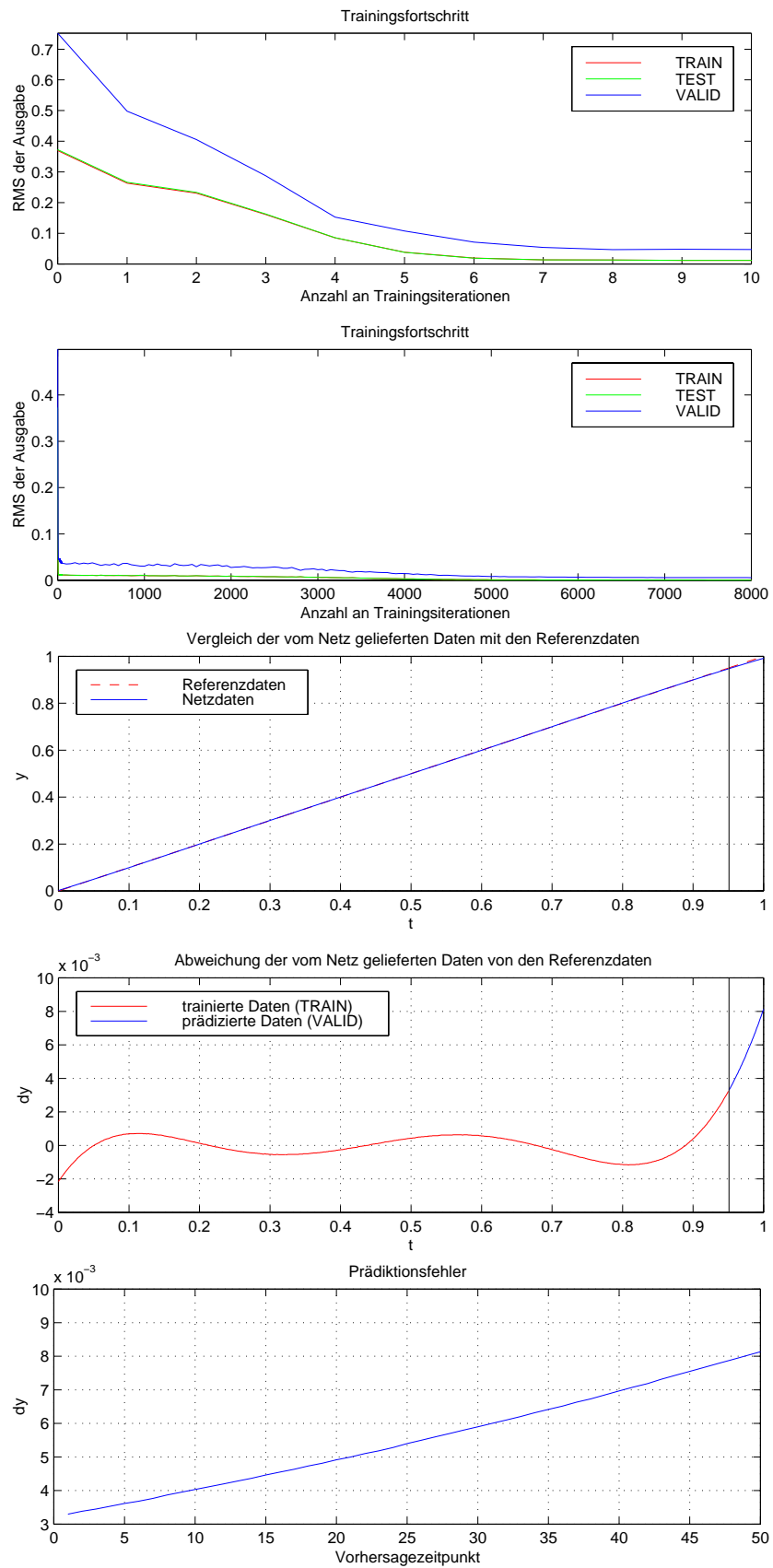


Abbildung 3.7: Trainingsfortschritt und Geradenapproximation Konfiguration 3

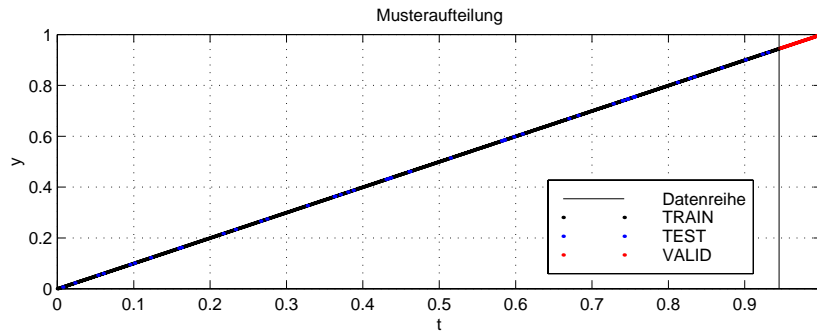


Abbildung 3.8: Musteraufteilung der Geraden

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	3	
Musteraufteilung	TRAIN+TEST: 95% davon TRAIN: 95% VALID: 5%	945 Muster 898 Muster 50 Muster
Eingabemuster	$y [1..3]$	$n=3$
Ausgabemuster	$y [1..3]$	$m=3$
Netzarchitektur	feedforward: 3-30-3	
Verknüpfung	vollständig	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.9$
Updatefunktion	Topological_Order	
Initialisierungsfunktion	Randomize_Weights	Intervall [-1; 1]
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	zufällig	

Die Musteraufteilung ist in Abbilung 3.8 gezeigt. Dabei wird stellvertretend für das gesamte Ein- und Ausgabemuster der erste Wert des Eingabemusters als Punkt dargestellt. Um das Ergebnis nach wiederum 8000 Trainingsiterationen zu beurteilen, können wir bei dieser Vorhersagestrategie für jeden Vorhersagezeitpunkt und jede Musterteilmenge einen RMS-Fehler berechnen. Dieser ist in den folgenden Tabellen in Einheiten von 10^{-3} zusammengestellt:

Vorhersagezeitpunkt	1	2	3	Mittel
$RM S_{TRAIN} \cdot 10^{-3}$	0.6456	0.6689	0.6748	0.6631
$RM S_{TEST} \cdot 10^{-3}$	0.6512	0.6770	0.6838	0.6707
$RM S_{VALID} \cdot 10^{-3}$	4.7981	5.0882	5.1629	5.0164

Anhand der Mittelwerte können nun wieder verschiedene Netzwerktopologien miteinander verglichen werden. Dazu variieren wir die Anzahl der versteckten Neuronen:

versteckte Neuronen	40	30	20	15	5
$RM S_{TRAIN} \cdot 10^{-3}$	1.003	0.6631	0.8424	0.7395	0.7649
$RM S_{TEST} \cdot 10^{-3}$	1.019	0.6707	0.8521	0.7612	0.7786
$RM S_{VALID} \cdot 10^{-3}$	7.3103	5.0164	5.9166	5.2795	5.0831

An dieser Stelle ist die maximale Anzahl der Neuronen gleichzeitig auch die optimale. Dies gilt allerdings nur für die aus Vergleichsgründen festgelegten 8000 Trainingsiterationen.

Um die Vorhersagequalität dieses Verfahrens grob einordnen zu können, ermitteln wir den durchschnittlichen Prädiktionsfehler aus Konfiguration 3. Dieser beträgt 5.4331, was anhand der untersten Graphik in Abbildung 3.7 leicht nachvollzogen werden kann. Da der RMS-Fehler der VALID-Muster hier nur geringfügig kleiner ist, können wir anhand dieses Beispiels wohl kaum eine Aussage über signifikante Qualitätsunterschiede zwischen den beiden Vorhersagestrategien machen.

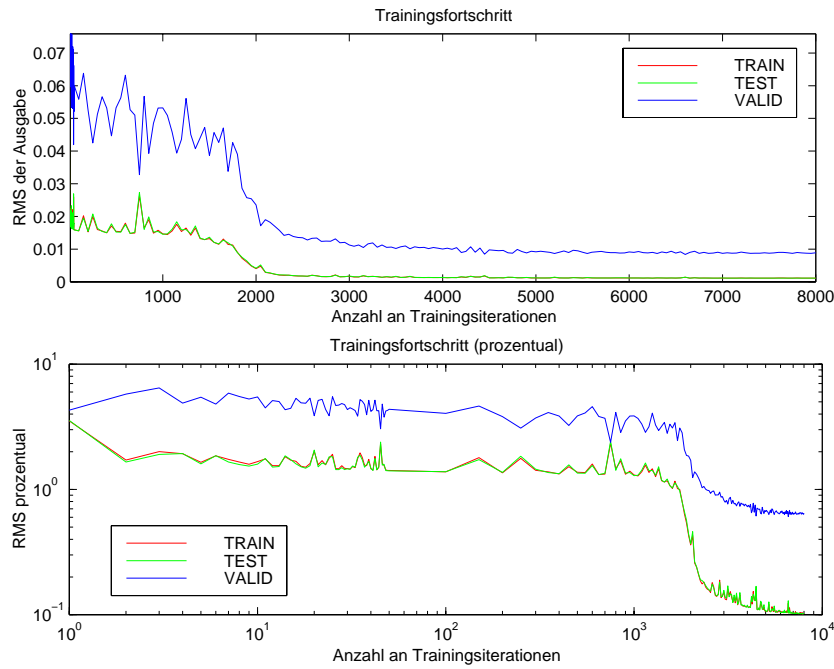


Abbildung 3.9: Trainingsfortschritt der Konfiguration 6

Die Entwicklung der Netzausgabefehler ist in Abbildung 3.9 dargestellt. Die untere Graphik zeigt den prozentualen Fehler im Vergleich zu den Ausgabefehlern nach der Netzinitialisierung in logarithmischer Skalierung. Der Fehler des Netzes nach der Initialisierung entspricht also 100 Prozent. Abbildung 3.10 zeigt die RMS-Fehler jeweils für die drei Vorhersagezeitpunkte und die Musterteilmengen. Deutlich ist der fast identische Verlauf von TRAIN- und TEST-Fehler zu beobachten. Der VALID-Fehler wächst mit zunehmendem Abstand vom letzten Trainingsmuster. Dies liegt daran, dass der Wertebereich der Trainingsmuster den der Validierungsmuster nicht enthält. Da sich jedoch die ersten Muster des Validierungsdatensatzes und die letzten Muster des Trainingsdatensatzes noch stark ähneln, entspricht der Fehler der ersten Validierungsmuster ungefähr dem der letzten Trainingsmuster. Mit steigender Nummer der Validierungsmuster wird der Offset zwischen den Eingabewerten der Validierungsmuster und denen der Trainingsmuster größer. Dieser Offset kann vom Netz nur sehr schlecht repräsentiert werden, da diese Information aufgrund der sich nicht überdeckenden Wertebereiche in den Trainingsdaten nicht enthalten ist.

Konfiguration 7

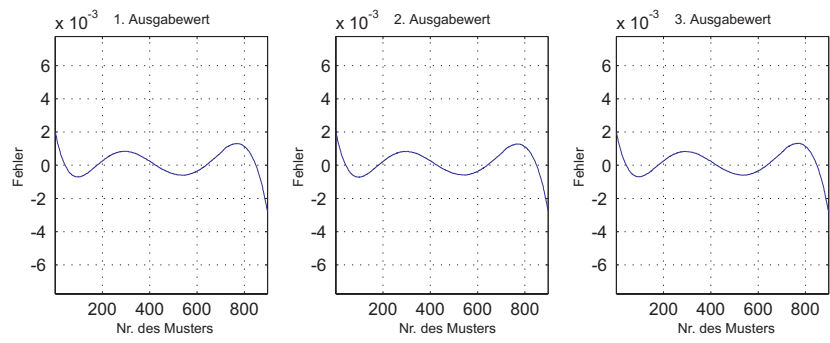
Zum Testen der Vorhersagestrategie 2 konstruieren wir ein neuronales Netz mit nur einem Ausgabeneuron. Ansonsten ist diese Konfiguration mit der vorhergehenden vergleichbar. Allerdings ändert sich mit der Anzahl der Ausgabeneuronen auch die Anzahl der maximal benötigten versteckten Neuronen auf 45.

Das Netz wird in einem ersten Schritt mit 8000 Iterationen trainiert. Der Verlauf der Fehlerkurven ändert sich nur geringfügig im Vergleich zur Konfiguration 6. Erstaunlicherweise sind die RMS-Fehler aller Musterteilmengen größer:

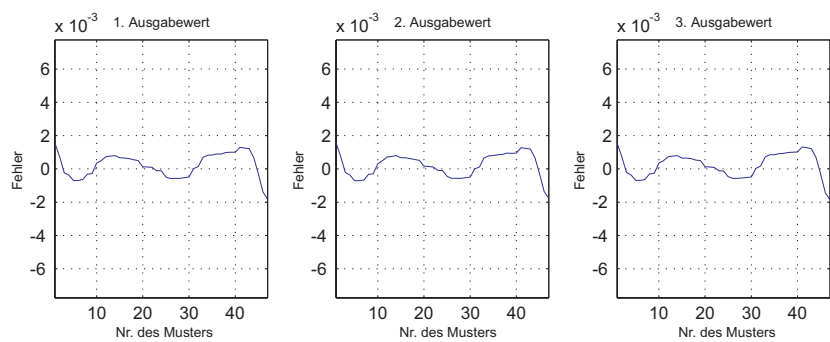
$RMS_{TRAIN} \cdot 10^{-3}$	1.0010
$RMS_{TEST} \cdot 10^{-3}$	1.0274
$RMS_{VALID} \cdot 10^{-3}$	7.2039

Auch durch Variation der Anzahl an versteckten Neuronen konnte das Ergebnis nicht verbessert werden.

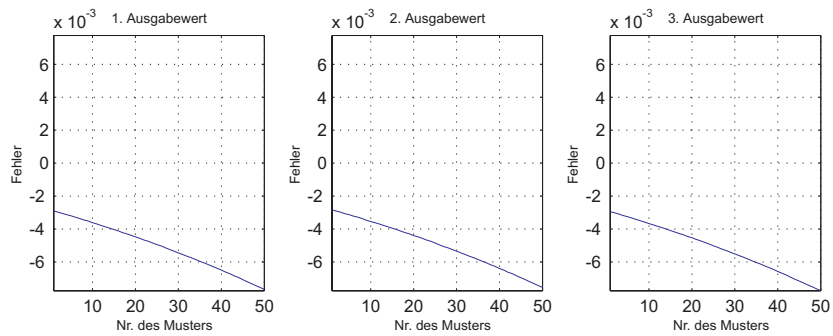
Der zweite Schritt besteht in der sukzessiven Berechnung der Vorhersagewerte. Da dieser Vorgang sowohl für den eigentlichen Vorhersagezeitraum als auch für die Trainings- und Testmuster geschehen kann, erhalten wir für alle drei Musterteilmengen RMS-Fehlerkurven in Abhängigkeit vom Vorhersagezeitpunkt (siehe Abbildung 3.11). Wie der Kurvenverlauf zeigt, ist die Qualität der Vorhersage nicht annähernd so gut wie die Resultate der Vorhersagestrategien 1 und 3. Die Fehler der Trainings- und Testmuster sind zwar noch annehmbar, der RMS-Fehler der Validierungsmuster, der direkt mit den Prädiktionsfehlern der Konfigurationen 1 bis 3 vergleichbar ist, erreicht allerdings einen um den Faktor 100 größeren Wert.



(a) TRAIN



(b) TEST



(c) VALID

Abbildung 3.10: Netzausgabefehler der drei Ausgabewerte für alle Muster

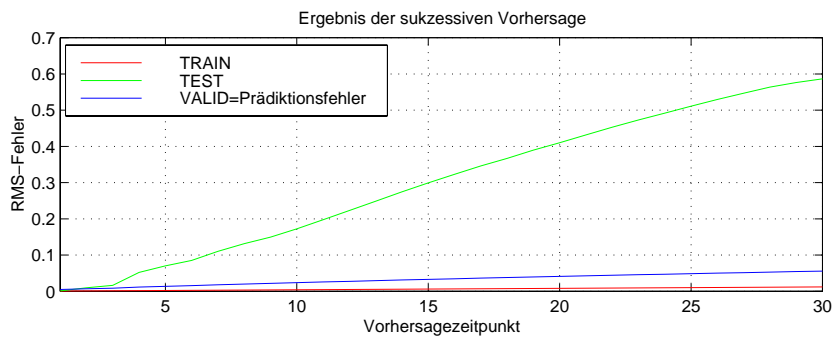


Abbildung 3.11: RMS-Fehler der Konfiguration 7

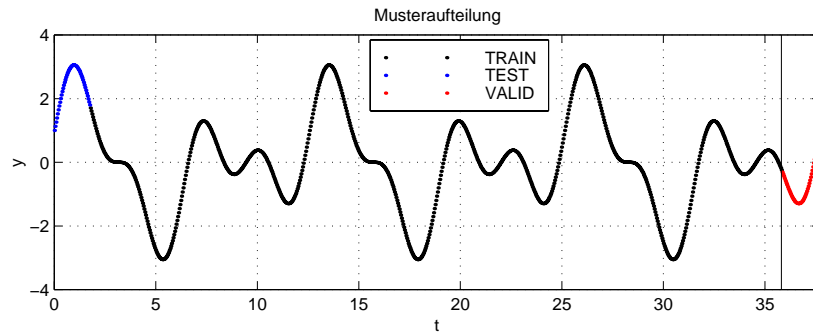


Abbildung 3.12: Musteraufteilung der periodischen Funktion

Die sich nicht überlappenden Wertebereiche wirken sich hier anscheinend noch stärker aus als in der letzten Konfiguration. Ein weiterer gravierender Nachteil ist wohl der sich bei dieser Strategie aufsummierende Fehler, da die bereits vorhergesagten fehlerbehafteten Werte als Ausgangswerte für weitere Vorhersagen dienen.

3.2.3 Vorhersage einer periodischen Funktion

Da wir bei der Vorhersage der Erdrotationsparameter davon ausgehen können, dass nach Abzug des linearen Anteils die zu prädizierenden Werte nicht oder nur gering außerhalb des Wertebereichs der vergangenen Daten liegen, konstruieren wir uns eine neue Funktion, die diese Voraussetzung erfüllt. Für diese Aufgabe bietet sich eine periodische Funktion an. Dazu berechnen wir uns die Werte der Funktion $y = \frac{3}{2}\sin(t) + \sin(2t) + \cos(\frac{t}{2})$ mit $t \in [0; 12\pi]$ wiederum an 1000 äquidistanten Stützstellen.

Die Musteraufteilung ist in Abbildung 3.12 dargestellt. Dass die Testmuster nun am Anfang der Datenreihe liegen, hat einen bestimmten Grund. Wie wir beim vorangegangenen Beispiel der Geraden feststellen konnten, entspricht der RMS-Fehler des Testdatensatzes überwiegend dem des Trainingsdatensatzes. Die Interpolationsfähigkeit des Netzes ist also sehr gut. Für den Zweck der Vorhersage allerdings ist die Fähigkeit zur Extrapolation wesentlich wichtiger. Um eine optimale Vorhersage zu erreichen, müsste das Training des Netzes beim kleinsten RMS-Fehler der Validierungsdaten abgebrochen werden, um ein *Overtraining* (Übertrainieren) zu vermeiden. Da uns die zukünftigen Daten als Referenzdaten in der Praxis jedoch nicht zur Verfügung stehen, können wir diesen Fehler nicht berechnen. Mit Hilfe eines kleinen Tricks umgehen wir das Problem: Für das neuronale Netz macht es keinen entscheidenden Unterschied, ob es in die Zukunft oder in die Vergangenheit extrapoliert. Deshalb können wir bei vorliegender Musteraufteilung annehmen, dass bei einem geringen Fehler der Testmuster auch der Fehler der Validierungsmuster klein ist.

Konfiguration 1

Zunächst soll Vorhersagestrategie 1 untersucht werden, d.h. unser Netz besitzt sowohl ein Eingabe- als auch ein Ausgabeneuron. Die folgenden Einstellungen liegen dieser Konfiguration zugrunde:

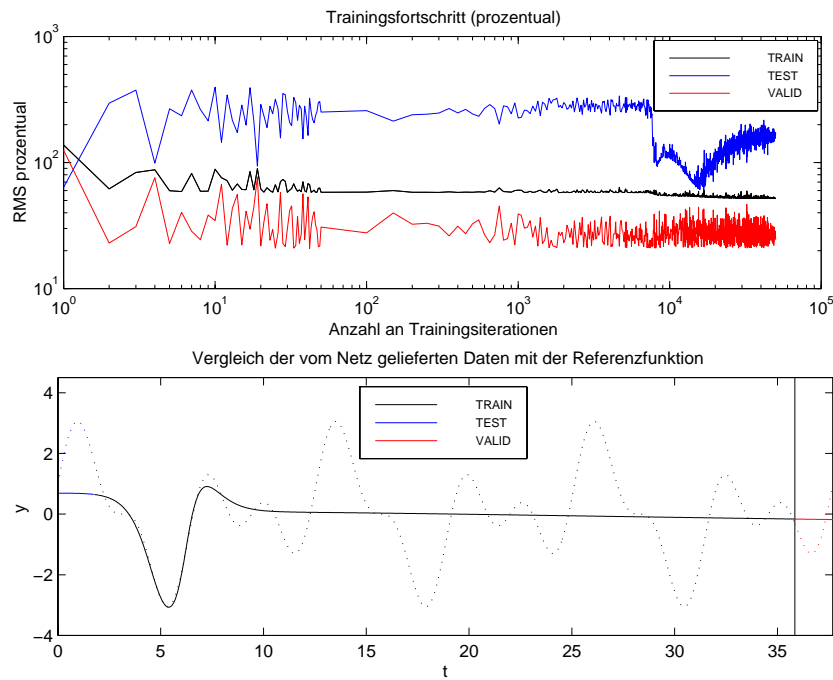


Abbildung 3.13: Lernfortschritt und Funktionsapproximation des Netzes 1-91-1

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	1	
Musteraufteilung	TRAIN+TEST: 95% davon TRAIN: 95% VALID: 5%	950 Muster 903 Muster 50 Muster
Eingabemuster	t	$n=1$
Ausgabemuster	y	$m=1$
Netzarchitektur	feedforward: 1-91-1	
Verknüpfung	vollständig	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.9$
Updatefunktion	Topological_Order	
Initialisierungsfunktion	Randomize_Weights	Intervall [-1; 1]
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	zufällig	

Trotz einer sehr langen Lernphase von 50 000 Trainingsiterationen verringert sich der Fehler des Trainingsdatensatzes nur geringfügig. Dementsprechend schlecht ist natürlich auch die Approximation der Funktion durch das Netz (siehe Abbildung 3.13). Die Netztopologie scheint für das vorliegende Problem also ungeeignet. Da die Näherungsformel zur Berechnung der maximal benötigten Neuronen nichts über deren topologische Verknüpfung aussagt, ist anzunehmen, dass ein Netz mit mehreren versteckten Schichten eventuell besser geeignet wäre. Dies widerspricht nur auf den ersten Blick der Aussage, jede stetige Funktion sei mit nur einer versteckten Schicht repräsentierbar. Würde das Netz weiter trainiert werden, so würde sich der Fehler bei richtiger Wahl der Lernrate kontinuierlich verkleinern, wenn auch nur in immer kleiner werdenden Schritten. Da dieses Vorgehen allerdings im Widerspruch zu einem möglichst ökonomischen Trainieren stehen würde, werden stattdessen verschiedene feedforward-Netztopologien gegenübergestellt und deren Lernerfolg jeweils anhand des RMS-Fehlers des Trainingsdatensatzes nach 8000 Iterationen qualifiziert:

Netztopologie	RMS_{TRAIN}	RMS_{TEST}	RMS_{VALID}
1-5-1	1.315	1.352	0.876
1-10-1	1.313	1.268	0.829
1-50-1	1.316	1.058	0.842
1-90-1	1.366	1.960	0.708
1-10-10-1	0.797	1.225	0.630
1-45-45-1	0.277	0.977	0.784
1-10-10-10-1	0.163	1.550	0.790
1-20-20-20-1	0.091	1.057	0.874
1-30-30-30-1	0.061	1.341	1.301

Wie wir gut erkennen können, hat der Fehler der Netze mit nur einer versteckten Schicht auch für verschiedenste Neuronenanzahlen einen konstanten und sehr hohen Wert. Erst durch Hinzunahme einer zweiten Schicht kommt es tatsächlich zu einer signifikanten Verbesserung der Lernsituation. Drei versteckte Schichten führen zu einer weiteren Reduzierung des Fehlers, der sich bei der Erhöhung der Neuronenanzahl pro Schicht nochmals verringert. In der Abbildung 3.14 sind die Ergebnisse einiger Netztopologien graphisch dargestellt. Die dreischichtigen Netze können zwar die Trainingsdaten sehr gut repräsentieren, an den Rändern des Trainingsdatensatzes nimmt der Fehler allerdings wie auch beim Beispiel der Geraden stark zu, sodass eine Extrapolation keine zufriedenstellenden Ergebnisse liefern kann. Der Netzausgabefehler des mit 8000 Iterationen trainierten Netzes mit drei versteckten Schichten und 30 Neuronen pro Schicht ist in Abbildung 3.15 zu sehen. Der durchschnittliche Prädiktionsfehler der Validierungsmuster, der beim späteren Vergleich mit den Strategien 2 und 3 eine wichtige Rolle spielen wird, beträgt hier 1.186. Auffällig ist vor allem auch die Tatsache, dass der Fehler innerhalb des Trainingsdatensatzes von links nach rechts zunimmt. Trotz zufälliger Präsentation der Trainingsmuster während des Lernens und Vertauschens der Musterreihenfolge innerhalb der PAT-Datei konnte dieser Effekt unerklärlicherweise nicht in seiner Auswirkung dezimiert werden.

Konfiguration 2

Die Unterschiede zur Konfiguration 1 sind:

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	3	
Musteraufteilung	TRAIN+TEST: 95% davon TRAIN: 95% VALID: 5%	937 Muster 890 Muster 49 Muster
Eingabemuster	y [1..5]	n=5
Ausgabemuster	y [1..3 5 10]	m=5
Netzarchitektur	feedforward, variabel	s.u.

Verschiedene Netztopologien werden mit Hilfe des mittleren RMS-Fehler nach 8000 Iterationen miteinander verglichen:

Netztopologie	RMS_{TRAIN}	RMS_{TEST}	RMS_{VALID}
5-18-5	0.0576	0.0344	0.0703
5-9-5	0.0569	0.0461	0.0725
5-9-9-5	0.0471	0.0277	0.0546

Die bereits sehr guten Ergebnisse lassen erahnen, dass diese Strategie die wohl am besten geeignete darstellt. Diese Annahme bestätigt sich, wenn wir den durchschnittlichen RMS-Fehler der Validierungsmuster von ca. 0.06 mit dem Prädiktionsfehler von 1.186 der Strategie 1 vergleichen. Auch wenn die beiden Fehler nicht direkt miteinander vergleichbar sind, so gibt es doch keine Zweifel über den Qualitätsunterschied beider Strategien.

Auch hier zeigt das Netz mit den zwei versteckten Schichten ein etwas besseres Lernverhalten. Die Prädiktions-eigenschaften dieses Netzes werden deshalb im folgenden genauer untersucht.

Abbildung 3.16 veranschaulicht den prozentualen Lernfortschritt über 50 000 Trainingsiterationen. Die RMS-Fehler besitzen nach Trainingsende nur noch ca. 0.4 Prozent des ursprünglichen Wertes nach der Netzinitialisierung. Die Fehlerkurven aller Musterteilmengen sind nahezu identisch. Dies liegt daran, dass sowohl die Testmuster als auch

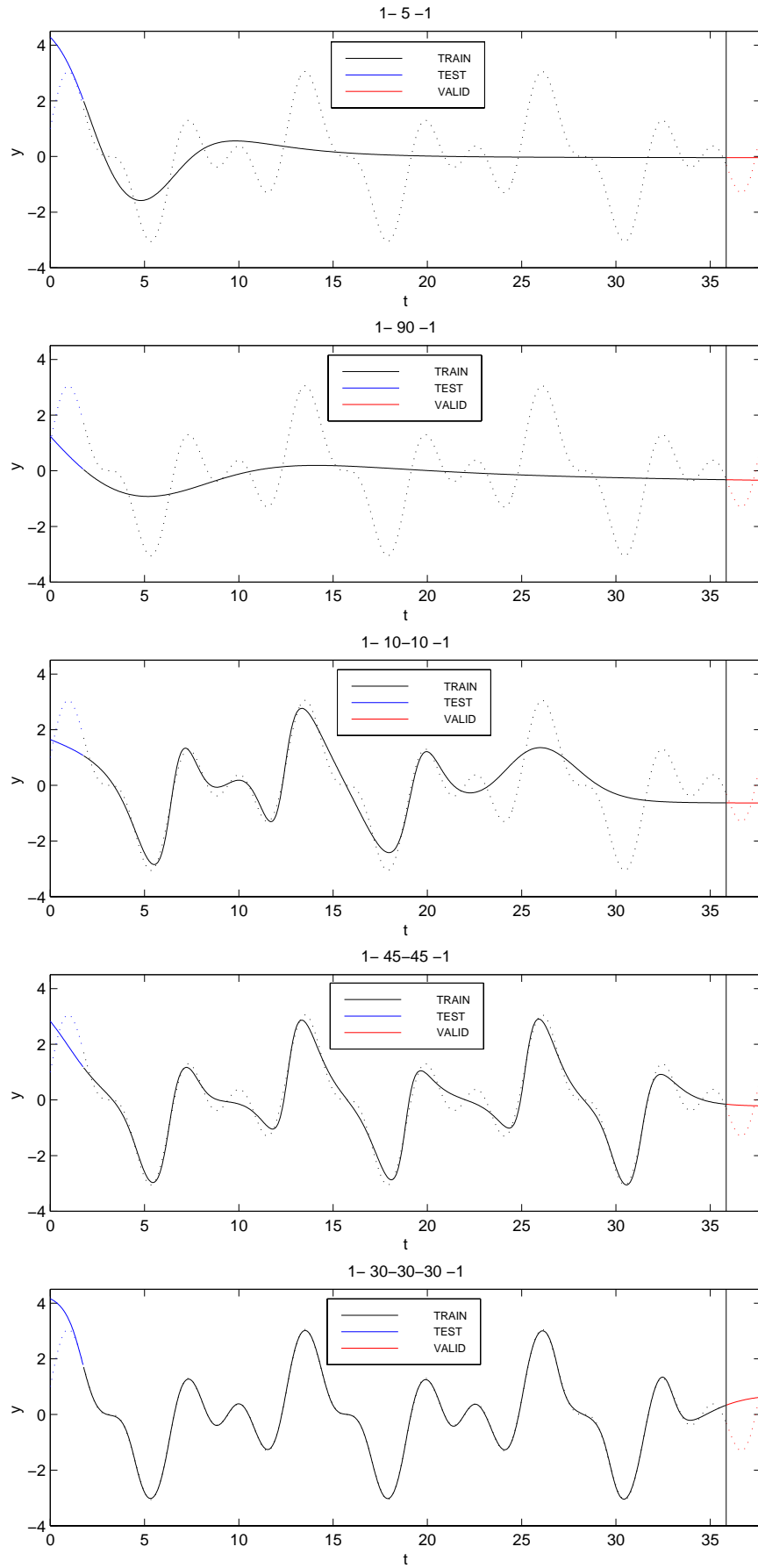


Abbildung 3.14: Vergleich der vom Netz gelieferten Daten mit der Referenzfunktion unterschiedlicher Topologien

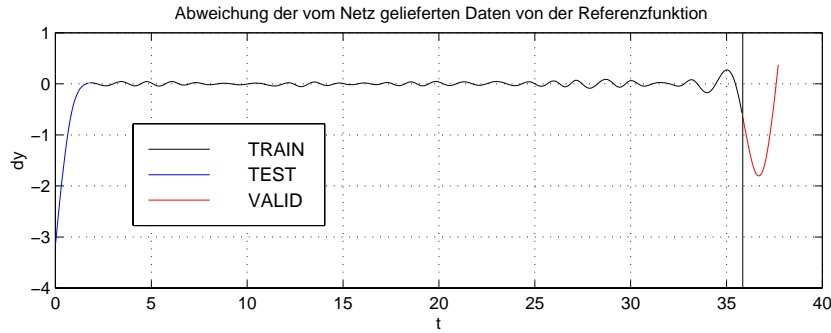


Abbildung 3.15: Ausgabefehler des Netzes 1-30-30-30-1 nach 50 000 Iterationen

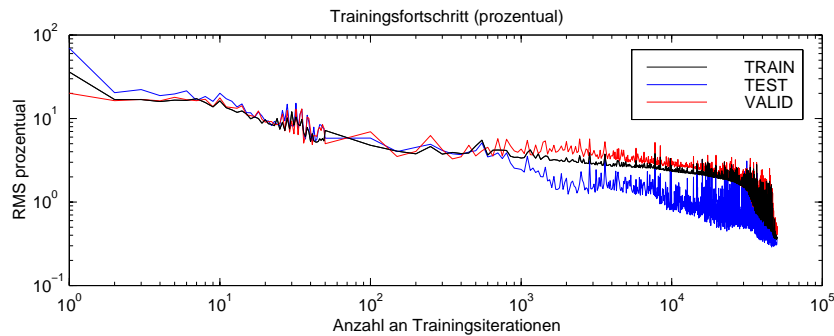


Abbildung 3.16: Lernfortschritt des Netzes 5-9-9-5

die Validierungsmuster aufgrund der Periodizität der Funktion im Trainingsdatensatz mit enthalten sind, sodass im vorliegenden Beispiel das Genauigkeitsmaß der Prädiktion mit dem der Repräsentation der Trainingsmuster weitgehend übereinstimmt.

Die Fehler aller Trainings-, Test- und Validierungsmuster zeigt Abbildung 3.17. Dabei werden die Ausgabefehler für alle fünf Ausgabeneuronen separat dargestellt. Der erste Ausgabewert entspricht dabei dem ersten Vorhersagezeitpunkt, der zweite dem zweiten, der dritte dem dritten, der vierte dem fünften und der fünfte dem zehnten. Dementsprechend groß ist der Fehler des fünften Ausgabeneurons im Vergleich zu den ersten vier. Zu beachten ist auch, dass der Fehler des ersten Ausgabeneurons größer erscheint als die Fehler des zweiten und dritten Neurons. Die Graphik der RMS-Fehler in Abbildung 3.18 für jedes Ausgabeneuron bestätigt dies. Offensichtlich hat das Netz Schwierigkeiten, den ersten Vorhersagewert zu bestimmen. Vermutlich ist der teilweise sehr kleine Unterschied der Funktionswerte zwischen dem letzten Eingabeneuron und dem ersten Ausgabeneuron der Grund für die unsichere Bestimmung.

Um sich ein Bild von der Vorhersagequalität machen zu können, werden abschließend in Abbildung 3.19 einige Musterpaare graphisch dargestellt.

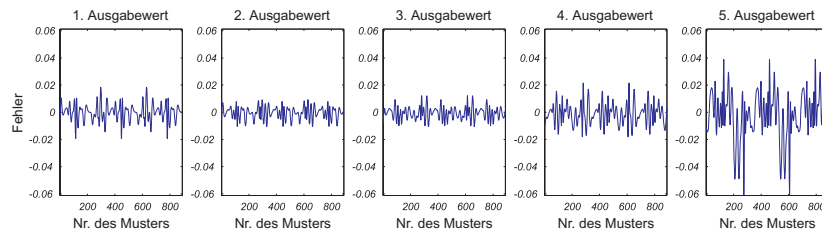
Konfiguration 3

Konfiguration 3 unterscheidet sich von Konfiguration 2 nur insofern, als nun Vorhersagestrategie 2 angewendet wird. Deshalb wird nur noch ein Ausgabeneuron benötigt. Um einen direkten Vergleich mit obigen Ergebnissen anstellen zu können, bleiben alle anderen Einstellmöglichkeiten unverändert.

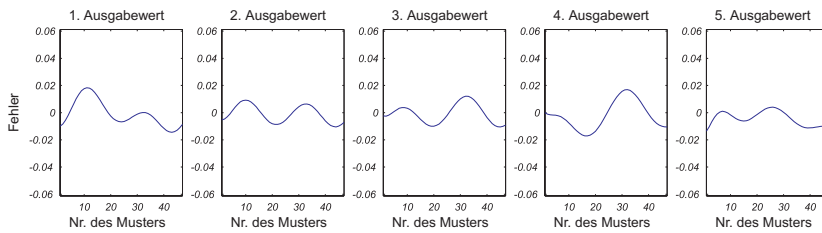
Nach dem Trainieren des Netzes ergibt sich ein RMS-Fehler der Trainingsdaten von 0.007. Dieser ist nur geringfügig größer als der Fehler des ersten Ausgabeneurons aus Konfiguration 2, der 0.006 beträgt.

Die anschließende sukzessive Vorhersage für die drei Musterdatensätze bringt abermals keine zufriedenstellenden Ergebnisse (siehe Abbildung 3.20). Ein Vergleich mit Vorhersagestrategie 3 zeigt den Qualitätsunterschied:

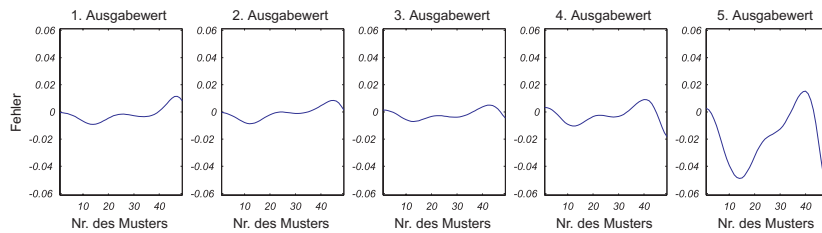
Trainingsmuster	
Vorhersagezeitpunkt	1 2 3 5 10
Vorhersagestrategie 2	0.007 0.015 0.026 0.057 0.175
Vorhersagestrategie 3	0.006 0.004 0.004 0.007 0.018



(a) Trainingsmuster



(b) Testmuster



(c) Validierungsmuster

Abbildung 3.17: Netzausgabefehler für jeweils alle Muster und Ausgabeneuronen

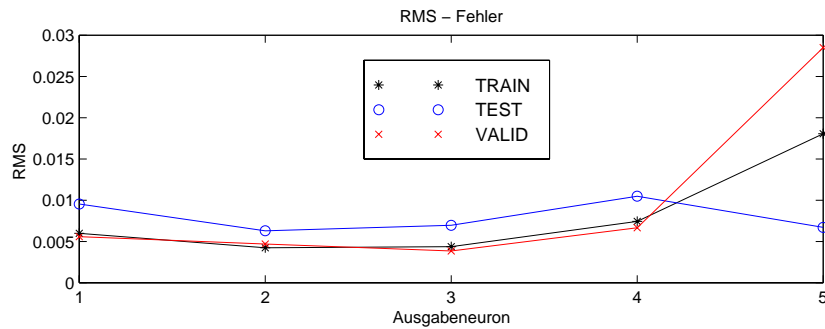
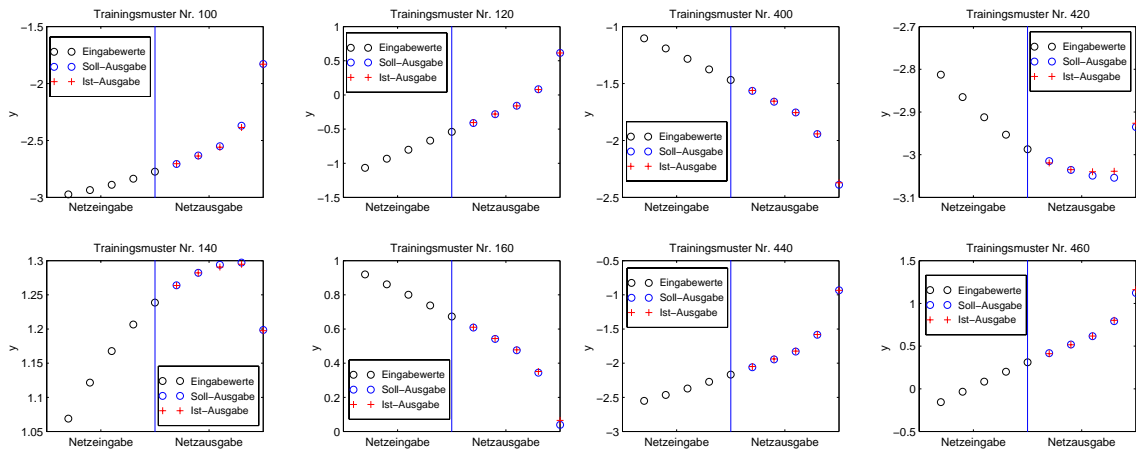
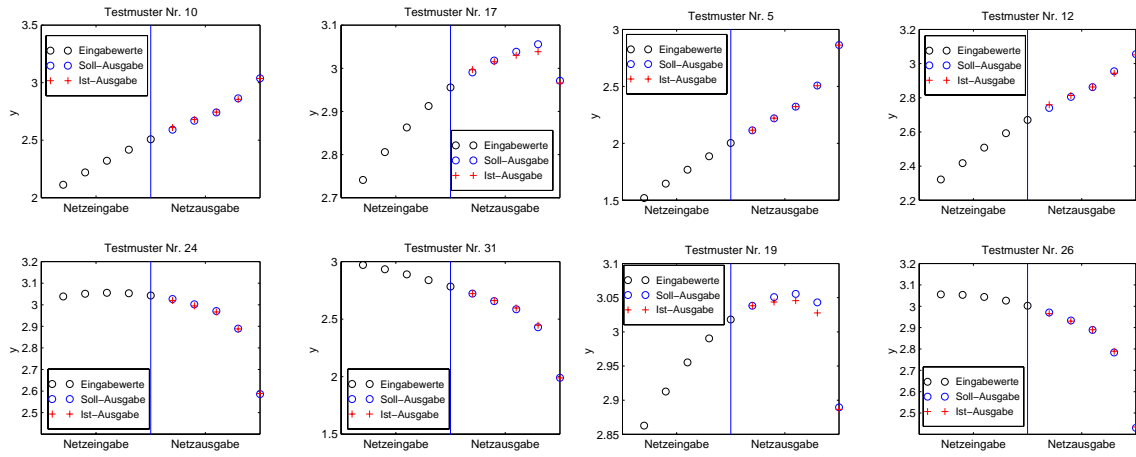


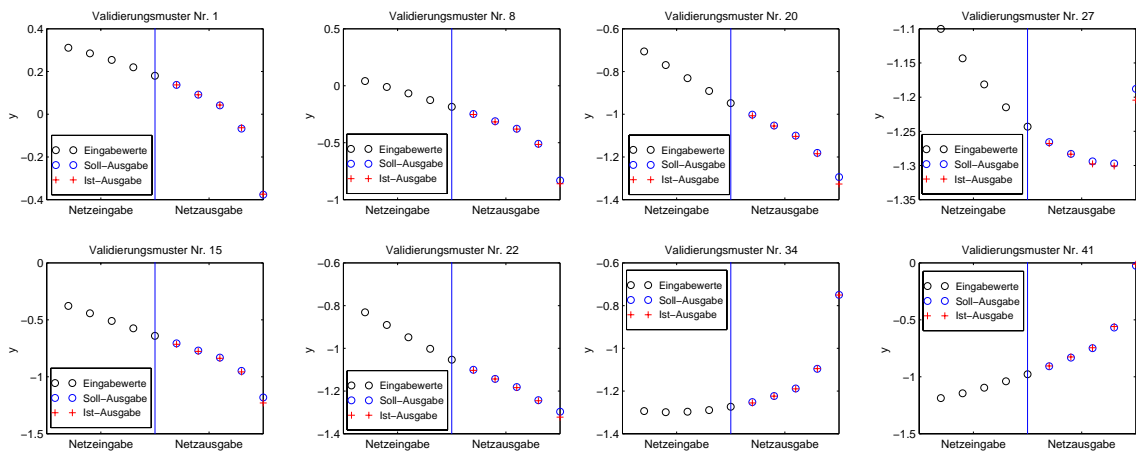
Abbildung 3.18: RMS-Fehler der fünf Ausgabeneuronen



(a) Trainingsmuster



(b) Testmuster



(c) Validierungsmuster

Abbildung 3.19: Darstellung einiger Musterpaare mit Netzein- und ausgabe

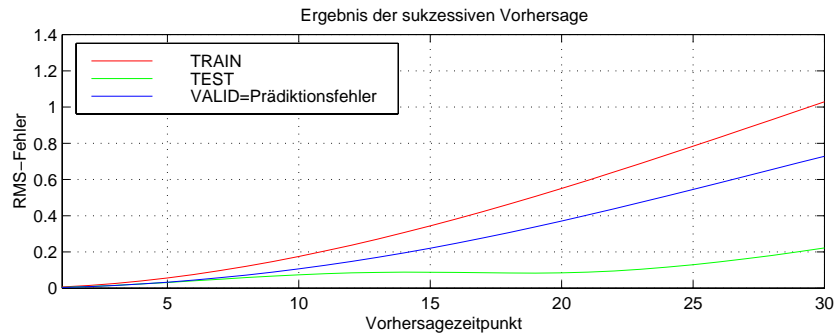


Abbildung 3.20: RMS-Fehler der Konfiguration 3

Testmuster					
Vorhersagezeitpunkt	1	2	3	5	10
Vorhersagestrategie 2	0.005	0.009	0.016	0.032	0.074
Vorhersagestrategie 3	0.009	0.006	0.007	0.010	0.008

Validierungsmuster					
Vorhersagezeitpunkt	1	2	3	5	10
Vorhersagestrategie 2	0.004	0.009	0.015	0.033	0.107
Vorhersagestrategie 3	0.005	0.005	0.004	0.007	0.028

Aufgrund der sehr guten Ergebnisse der Vorhersagestrategie 3 wird diese für die Prädiktion der Erdrotationsparameter gewählt.

Konfiguration 4

Abschließend sollen weitere Netzwerkarchitekturen untersucht werden. Dazu wenden wir die Einstellungen der Konfiguration 3 jeweils auf ein Jordan- und ein Elman-Netz an, die die gleiche Anzahl an Eingabe-, Ausgabe- und versteckten Neuronen besitzen, wie das entsprechende feedforward-Netz. Die versteckte Schicht wird durch Kontextzellen ergänzt. Die Architektur der Jordan- und Elman-Netze erfordert die Anwendung charakteristischer Update- und Initialisierungsfunktionen. Außerdem ist es wichtig, die Musterpräsentation während des Lernens in chronologischer Reihenfolge vorzunehmen, da sonst das Netz nicht den vergangenen Ausgabewert im "Gedächtnis" behält und diesen als zusätzliche Eingabeinformation nutzt, sondern einen beliebigen. Es folgen die Einstellungen im Detail:

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	3	
Musteraufteilung	TRAIN+TEST: 95% davon TRAIN: 95% VALID: 5%	937 Muster 890 Muster 49 Muster
Eingabemuster	$y [1..5]$	$n=5$
Ausgabemuster	$y [1..3 \ 5 \ 10]$	$m=5$
Netzarchitektur	Jordan: 5-9-9-5 Elman: 5-9-9-5	
Verknüpfung	vollständig	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.9$
Updatefunktion	JE_Order JE_Special	
Initialisierungsfunktion	JE_Weights	Intervall [-1; 1]
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	chronologisch	

Die Ergebnisse und der Vergleich mit Konfiguration 2 zeigt, dass die partiell rekurrenten Netzwerke unter Verwendung der Vorhersagestrategie 3 keinerlei Verbesserungen bewirken:

<i>RMS_{TRAIN}</i>					
Vorhersagezeitpunkt	1	2	3	5	10
feedforward	0.006	0.004	0.004	0.007	0.018
Jordan	0.006	0.005	0.004	0.009	0.022
Elman	0.006	0.005	0.006	0.010	0.019

<i>RMS_{TEST}</i>					
Vorhersagezeitpunkt	1	2	3	5	10
feedforward	0.009	0.006	0.007	0.010	0.008
Jordan	0.010	0.009	0.007	0.013	0.018
Elman	0.009	0.010	0.008	0.013	0.018

<i>RMS_{VALID}</i>					
Vorhersagezeitpunkt	1	2	3	5	10
feedforward	0.005	0.005	0.004	0.007	0.028
Jordan	0.004	0.008	0.005	0.011	0.031
Elman	0.004	0.008	0.004	0.010	0.029

Aufgrund der teilweise sogar schlechteren Ergebnisse wird auf ihre Anwendung zur Vorhersage der Erdrotationsparameter verzichtet.

Kapitel 4

Analyse und Reduktion der Zeitreihen

4.1 Erdrotation (ΔLOD)

In diesem Kapitel werden Analyse und Reduktion der Zeitreihe C04 des IERS exemplarisch für den Erdrotationsparameter ΔLOD detailliert geschildert. In den sich anschließenden Kapiteln 4.2 bzw. 4.3 werden dann entsprechend für ΔUT1 und die Polbewegung lediglich spezifische Charakteristika und die Ergebnisse illustriert.

4.1.1 Darstellung der Zeitreihe und Frequenzanalyse

Grundlage für die Analyse ist die C04-Reihe des IERS, die die Erdrotationsparameter von 1962 bis heute im Tagesabstand enthält. Wir nutzen die Daten im Zeitraum vom 1.1.1962 bis einschließlich 31.12.1998 für unsere Zwecke. Abbildung 4.1 zeigt ΔLOD in diesem Zeitraum und das dazugehörige Amplitudenspektrum. Deutlich sind im Spektrum die jährliche und halbjährliche Periode zu erkennen. Desweiteren fallen relativ hohe Spitzen im hochfrequenten Bereich auf, mit Perioden unter 50 Tagen. Die Ursache dafür ist der Einfluss der Gezeiten der festen Erde und der Meeresgezeiten. Die Periode von ca. 121 Tagen ist dagegen der Atmosphäre zuzuschreiben. Ziel ist es nun, die bekannten Einflüsse der Gezeiten und der saisonalen Schwankungen zu modellieren und von den Beobachtungen zu subtrahieren.

4.1.2 Gezeitenmodellierung

Das hier verwendete Gezeitenmodell ist, wie bereits in Kapitel 2.1.2 bemerkt, von Yoder et al. (1981) entwickelt worden. Es beinhaltet die Deformationen der Erde unter der Annahme eines entkoppelten Kerns, dargestellt mit Hilfe einer Reihenentwicklung. Die Reihenkoeffizienten in Tabellenform und Hinweise zur Berechnung sind im IERS Annual Report (1999) und in McCarthy (1996) angegeben. Der Gesamteffekt der Gezeiten wird dabei in mehrere Teileffekte zerlegt. So existieren für die Perioden von 5 Tagen bis 35 Tagen ($UT1R$, $LODR$ und ωR), für die Perioden von 5 Tagen bis 18.6 Jahren ($UT1S$, $LODS$ und ωS) und für die eintägigen und halbtägigen Perioden ($UT1D$, $LODD$, ωD) separate Tabellen. Der Aufbau ist für jede dieser Tabellen leicht unterschiedlich. Auch die Anzahl der enthaltenen Gezeitenperioden ist nicht identisch. Daher soll jeweils in einem ersten Schritt auf den Tabellenaufbau eingegangen werden. Anschließend wird dann die Berechnung der Gezeitenkorrekturen am Beispiel von LOD erläutert.

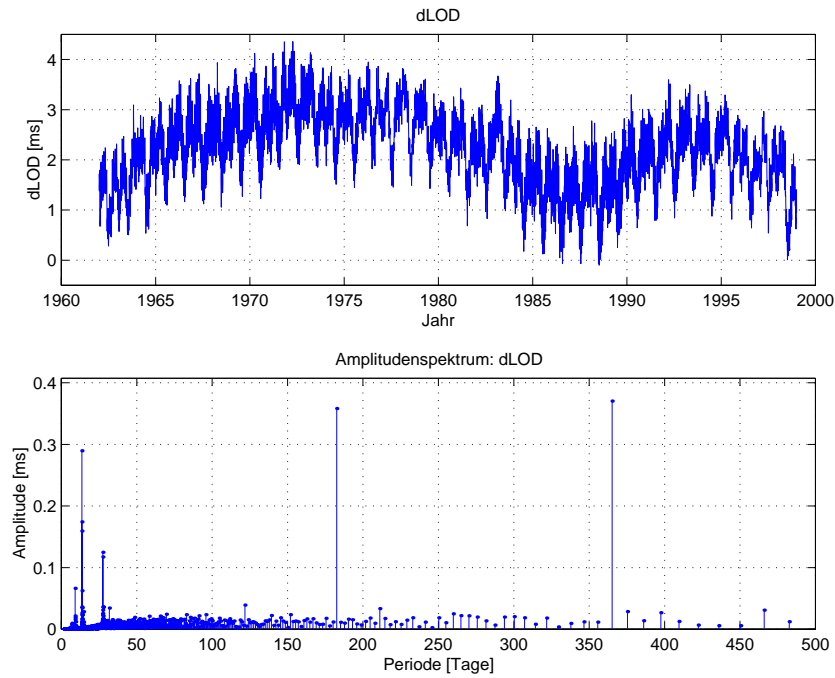


Abbildung 4.1: Δ LOD und Spektrum

Einfluss der zonalen Gezeiten mit Perioden von 5 bis 35 Tagen¹

Argument					Periode Tage	UT1 – UT1R 10^{-4} s	Δ LOD – Δ LODR 10^{-5} s
l	l'	F	D	Ω			
1	0	2	2	2	5.64	-0.02	0.3
2	0	2	0	1	6.85	-0.04	0.4
					⋮		
-1	-1	0	2	0	34.85	-0.09	0.2

Die ersten fünf Spalten beinhalten die Multiplikatoren der sog. *Astronomischen Fundamentalargumente* für insgesamt 41 Perioden, die sich von 5.64 Tagen bis zu 34.85 Tagen erstrecken und in der sechsten Spalte in aufsteigender Reihenfolge zu finden sind. In den folgenden Spalten sind jeweils für die drei zu korrigierenden Zeitreihen die Koeffizienten für die Sinus- bzw. Cosinusterme angegeben.

Für die Modellierung der Gezeiten ist es also zunächst notwendig, die Fundamentalargumente für den jeweiligen Berechnungszeitpunkt zu bestimmen:

Argument	Bedeutung Berechnung
l	Mittlere Anomalie des Mondes $=134.96340251^\circ + 1717915923.2178'' t + 31.8792'' t^2 + 0.051635'' t^3 - 0.00024470'' t^4$
l'	Mittlere Anomalie der Sonne $=357.52910918^\circ + 129596581.0481'' t - 0.5532'' t^2 + 0.000136'' t^3 - 0.00001149'' t^4$
F	(Mittlere Länge des Mondes) - Ω $=93.27209062^\circ + 1739527262.8478'' t - 12.7512'' t^2 - 0.001037'' t^3 + 0.00000417'' t^4$
D	Mittlere Breite des Mondes relativ zur Sonne $=297.85019547^\circ + 1602961601.2090'' t - 6.3706'' t^2 + 0.006593'' t^3 - 0.00003169'' t^4$
Ω	Mittlere Länge des aufsteigenden Bahnknotens des Mondes $=125.04455501^\circ - 6962890.2665'' t + 7.4722'' t^2 + 0.007702'' t^3 - 0.00005939'' t^4$

¹Die Korrektur $\omega - \omega R$ wird in dieser Arbeit nicht verwendet und deshalb in den Tabellen nicht mit abgedruckt.

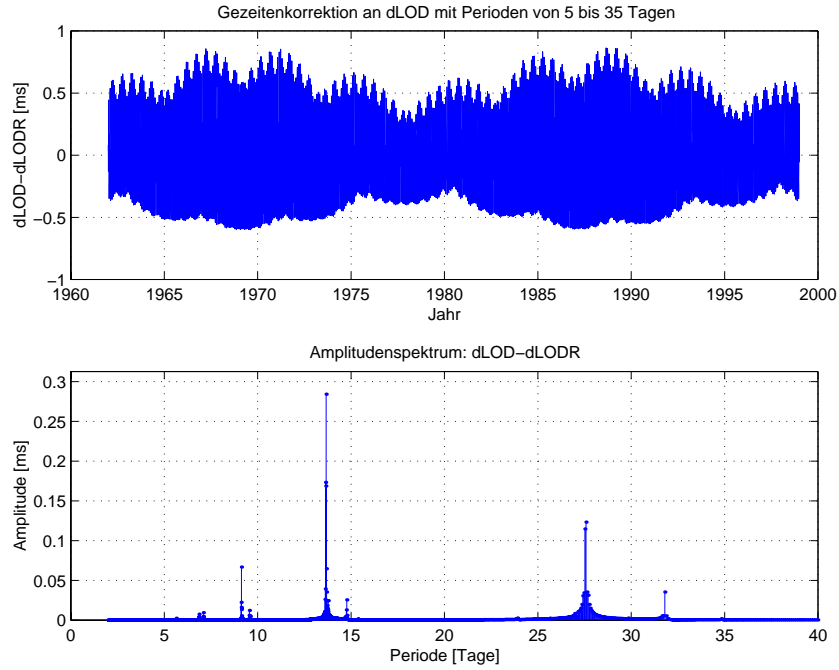


Abbildung 4.2: Gezeitenkorrektion für ΔLOD mit Perioden von 5 bis 36 Tagen

Die Variable t muss hier in Julianischen Jahrhunderten zu 36 525 Tagen à 86 400 Sekunden bezüglich TAI angegeben werden. Der Nullpunkt liegt bei J2000.0 (vgl. Kapitel 2.1.1). Die Berechnung von t z.B. aus MJD erfolgt mit

$$t = \frac{JD - 2\,451\,545}{36\,525} = \frac{(MJD + 2\,400\,000.5) - 2\,451\,545}{36\,525} = \frac{MJD - 51\,544.5}{36\,525}. \quad (4.1)$$

Die fünf berechneten Fundamentalarargumente α_j (l, l', F, D, Ω) werden anschließend mit den entsprechenden Multiplikatoren a_{ij} aus den ersten fünf Spalten der obigen Tabelle multipliziert und für jede Gezeiten-Periode i aufsummiert:

$$\xi_i = \sum_{j=1}^5 a_{ij} \alpha_j \quad (4.2)$$

Um die eigentliche Korrektion $\Delta LOD - \Delta LODR$ zu erhalten, müssen wir noch den Cosinus von ξ_i mit dem dazugehörigen Cosinus-Koeffizienten C_{LOD} aus Spalte 8 multiplizieren und das Ergebnis für alle Perioden aufsummieren:

$$\Delta LOD - \Delta LODR = \sum_{i=1}^{41} C_{LOD} \cdot \cos \xi_i \quad (4.3)$$

Diese Korrektion können wir nun für alle Zeitpunkte unserer Datenreihe ΔLOD berechnen und erhalten somit die um die zonalen Gezeiten mit Perioden bis zu 35 Tagen reduzierten Werte $\Delta LODR$:

$$\Delta LODR = \Delta LOD - (\Delta LOD - \Delta LODR) \quad (4.4)$$

Abbildung 4.4 zeigt die Ergebnisse für das Berechnungsintervall (1962-1998) und das dazugehörige Spektrum. Große Cosinus-Koeffizienten entsprechen im Spektrum großen Amplituden. Zum Vergleich sind deshalb die acht Perioden mit den größten Cosinus-Koeffizienten zusammengestellt:

Cos-Koef. 10^{-5}s	35.7	18.8	14.8	6.8	3.6	3.1	2.8	1.5
Periode Tage	13.66	27.56	13.63	9.13	31.81	14.77	9.12	13.78

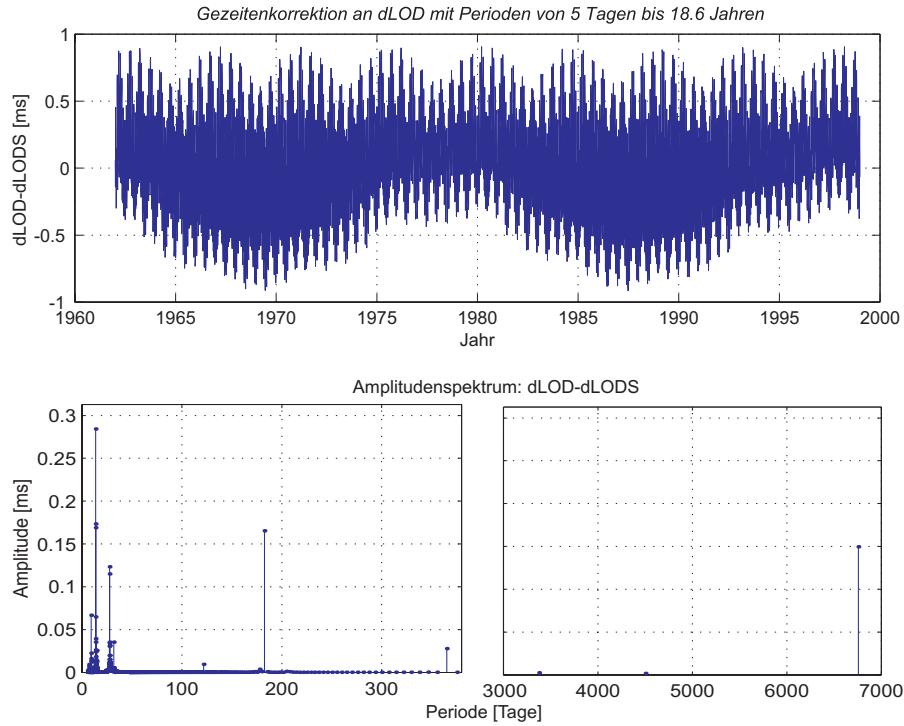


Abbildung 4.3: Gezeitenkorrektur für ΔLOD mit Perioden von 5 Tagen bis 18.6 Jahren

Einfluss der zonalen Gezeiten und der Meeresgezeiten mit Perioden von 5 Tagen bis 18.6 Jahren

Argument					Periode	UT1 – UT1S		$\Delta LOD - \Delta LODS$	
l	l'	F	D	Ω		S_{UT1} 10^{-4} s	C_{UT1} 10^{-4} s	C_{LOD} 10^{-5} s	S_{LOD} 10^{-5} s
1	0	2	2	2	5.64	-0.02	-	0.3	-
2	0	2	0	1	6.85	-0.04	-	0.4	-
					⋮				
0	0	0	0	1	6798.38	-1637.68	-0.10	-15.2	0.0

Diese Tabelle beinhaltet neben den schon bekannten 41 Perioden noch 21 zusätzliche Schwingungen im Bereich von 35 Tagen bis 18.6 Jahren. Ferner wird für den gesamten Spektralbereich auch der Einfluss der Meeresgezeiten berücksichtigt. Letzterer wird bei ΔLOD durch die zusätzlichen Sinus-Koeffizienten S_{LOD} in der letzten Spalte zum Ausdruck gebracht. Es besteht also ein Phasenversatz von $\frac{\pi}{2}$ zwischen den zonalen Gezeiten der festen Erde und den Ozeangezeiten. Die Sinus-Koeffizienten der Ozeangezeiten sind allerdings betragsmäßig höchstens $\frac{1}{10}$ so groß wie die Cosinus-Koeffizienten der Erdgezeiten und spielen daher nur eine untergeordnete Rolle.

Die Berechnung der Korrektur erfolgt analog:

$$\Delta LOD - \Delta LODS = \sum_{i=1}^{62} C_{LOD} \cdot \cos \xi_i + S_{LOD} \cdot \sin \xi_i \tag{4.5}$$

Zum Vergleich der Ergebnisse in Abbildung 4.3 dient folgende Tabelle der größten Amplituden²:

Cos-Koef. 10^{-5} s	-1637.68	-48.84	35.7	18.8	-15.55	7.9	6.8	3.6
Periode Tage	6798.38	182.62	13.66	27.56	365.26	3399.18	9.13	31.81

²Die Periode von 3399.18 Tagen entspricht ca. 9.3 Jahren und damit der Hälfte der durch die Mondbahn verursachten Nutationsperiode von 18.6 Jahren.

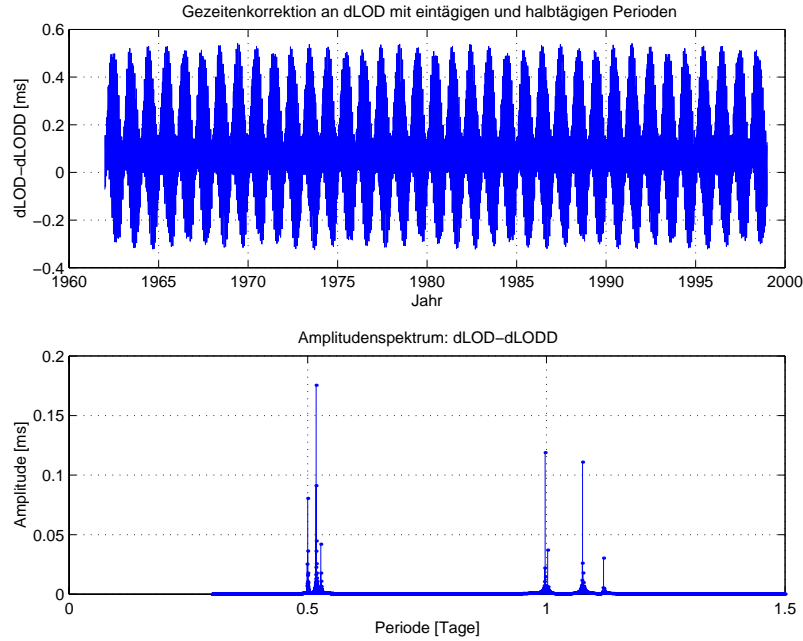


Abbildung 4.4: Korrektur der Meereszeiten für ΔLOD mit eintägigen und halbtägigen Perioden

Einfluss der eintägigen und halbtägigen Ozeangezeiten

Argument						Phase	Periode	UT1 – UT1D		$\Delta LOD - \Delta LODD$	
l	l'	F	D	Ω	θ	ϕ Grad	Tage	S_{UT1} 10^{-4} s	C_{UT1} 10^{-4} s	C_{LOD} 10^{-5} s	S_{LOD} 10^{-5} s
-1	0	-2	0	-2	1	-90	26.868	0.02	0.05	-1.4	2.8
0	0	-2	0	-2	1	-90	25.819	0.12	0.16	-7.1	9.4
							⋮				
0	0	0	0	0	2	0	11.967	-0.02	0.00	2.5	-0.5

In dieser Tabelle sind die Koeffizienten der Auswirkung der Meereszeiten mit eintägigen und halbtägigen Perioden verzeichnet. Diese gehen auf die Veröffentlichung von Ray et al. (1995) zurück. Vergleichsweise wurden auch die noch genaueren von Gipson (1996) angegebenen Koeffizienten verwendet. Der Unterschied zwischen diesen beiden Berechnungsgrundlagen ist für unsere Zwecke allerdings nicht signifikant.

Auffällig sind die zwei zusätzlichen Spalten: der Erdrotationswinkel θ und die Phase ϕ . θ ist, wie die Fundamentalargumente selbst, eine Funktion der Zeit:

$$\theta = 100.4606184^\circ + 360.98564737^\circ (MJD - 51544.5) \tag{4.6}$$

Das Argument ξ_i kann dann für jede Periode analog zur Gleichung 4.2 berechnet werden (Haas, 1995):

$$\xi_i = \sum_{j=1}^6 c_{ij} \gamma_j + \phi_i \tag{4.7}$$

Hier sind c_{ij} die jeweiligen Multiplikatoren der nun sechs Argumente γ_i ($l, l', F, D, \Omega, \theta$) für die i . Periode. Entsprechend Gleichung 4.5 ist dann

$$\Delta LOD - \Delta LODD = \sum_{i=1}^8 C_{LOD} \cdot \cos \xi_i + S_{LOD} \cdot \sin \xi_i. \tag{4.8}$$

Die Korrektur und ihr Spektrum sind in Abbildung 4.4 zu sehen.

Aufgrund der zu geringen Auflösung der Datenreihen von nur einem Tag sind diese hochfrequenten Anteile allerdings nicht mehr enthalten. Da jedoch die Frequenzen der eintägigen und halbtägigen Schwingungen jeweils sehr nahe

beieinander liegen, treten langperiodische Schwabungen auf, die dann berücksichtigt werden müssten, wenn die Korrektion nur unvollständig gelänge. Eine Analyse des Spektrums des bereits korrigierten ΔLOD s führte jedoch zum Ergebnis, dass darin keine Schwabungen mehr enthalten sind und sich damit eine Reduktion erübrigt.

4.1.3 Saisonale Einflüsse und linearer Anteil

Neben dem Einfluss der Gezeiten auf ΔLOD ist auch die Tatsache bekannt, dass saisonale atmosphärische Variationen auf die Erdrotation einwirken. Diese sind leider im Gegensatz zu den Gezeiten nicht exakt modellierbar. Da uns aber die Dauer eines Jahres bekannt ist, können wir die Parameter der jährlichen und halbjährlichen Schwingung mittels Ausgleichung schätzen. Außerdem erweist es sich als sinnvoll, Offset und Steigung eines linearen Ansatzes mitzuschätzen. Durch diese beiden zusätzlichen Freiheitsgrade können wir vermeiden, dass das Modell der Wirklichkeit nicht genau genug entspricht und somit die Ausgleichung nicht konvergiert. Sind die Parameter geschätzt, kann die Zeitreihe neben den Gezeiten nun auch um die saisonalen Komponenten und den linearen Anteil bereinigt werden.

Abgesehen davon, ist die Analyse der Variation von jährlicher und halbjährlicher Schwingung sowie des linearen Anteils für die Vorhersage unerlässlich. Die Ergebnisse geben uns einen Anhaltspunkt, in welchem Maß sich diese bekannten Komponenten von ΔLOD innerhalb des Prädiktionszeitraums verändern. Wichtig sind diese Erkenntnisse vor allem für die Vorhersage der kurzfristigen Schwankungen. Hierbei werden im Vorfeld neben dem zu jedem Zeitpunkt bekannten Einfluss der Gezeiten auch diese Komponenten subtrahiert. Auf die eigentlich vorhergesagten Residuen werden sie anschließend wieder addiert, um Vorhersagen für ΔLOD zu erhalten.

Die Komponenten sollen mit Hilfe einer vermittelnden Ausgleichung nach der Methode der kleinsten Quadrate (Parameterschätzung im Gauß-Markov-Modell) geschätzt werden (Ebner, 1998).

Den linearen Ansatz, der Offset und Steigung enthält, können wir durch $\Delta\text{LOD}_{lin} = a_0 + a_1 t$ ausdrücken. Der Parameter a_0 entspricht dem Offset, a_1 repräsentiert die Steigung. Die saisonalen Komponenten stellen wir jeweils als Sinus-Schwingung $\Delta\text{LOD}_{j/hj} = A \sin(2\pi f t + \phi)$ dar. Die zu bestimmenden Parameter sind hier die Amplitude A , die Frequenz f und die Phase ϕ . Da wir auch die Entwicklung der Phase analysieren möchten, ziehen wir diese Darstellung der üblichen Fourier-Darstellung mit Sinus- und Cosinus-Term vor, obwohl eine Umrechnung der beiden Fourierkoeffizienten in die Phase einfach durchzuführen ist.

Wahl des mathematischen Modells

Das Funktionalmodell können wir damit als Summe dieser Einflüsse durch folgende Formel beschreiben:

$$\Delta\text{LOD}_{lin,j,hj} = a_0 + a_1 t + A_j \sin(2\pi f_j t + \phi_j) + A_{hj} \sin(2\pi f_{hj} t + \phi_{hj}) \quad (4.9)$$

Beim Aufstellen des stochastischen Modells wählen wir für die Gewichtskoeffizientenmatrix Q_{bb} der Beobachtungen die Einheitsmatrix. Dies ist für unsere Zielsetzungen der Datenanalyse und -reduktion ausreichend. Damit ist auch die Gewichtsmatrix P_{bb} , die als Inverse der Gewichtskoeffizientenmatrix definiert ist, die Einheitsmatrix.

Durchführung der Ausgleichung

Da es sich um ein nicht-lineares Problem handelt, müssen wir zunächst linearisieren. Dazu benötigen wir die partiellen Ableitungen von ΔLOD_{ljh} ³ nach den insgesamt $u = 8$ unbekanntem Parametern a_0 , a_1 , A_j , f_j , ϕ_j , A_{hj} , f_{hj} und ϕ_{hj} :

- $\frac{\partial \Delta\text{LOD}_{ljh}}{\partial a_0} = 1$
- $\frac{\partial \Delta\text{LOD}_{ljh}}{\partial a_1} = t$
- $\frac{\partial \Delta\text{LOD}_{ljh}}{\partial A_j} = \sin(2\pi f_j t + \phi_j)$
- $\frac{\partial \Delta\text{LOD}_{ljh}}{\partial f_j} = 2A_j \pi t \cos(2\pi f_j t + \phi_j)$
- $\frac{\partial \Delta\text{LOD}_{ljh}}{\partial \phi_j} = A_j \cos(2\pi f_j t + \phi_j)$

³Aus Gründen der Übersichtlichkeit wird der Index "lin,j,hj" durch "ljh" ersetzt.

- $\frac{\partial \Delta LOD_{l_j h}}{\partial A_{h_j}} = \sin(2\pi f_{h_j} t + \phi_{h_j})$
- $\frac{\partial \Delta LOD_{l_j h}}{\partial f_{h_j}} = 2A_{h_j} \pi t \cos(2\pi f_{h_j} t + \phi_{h_j})$
- $\frac{\partial \Delta LOD_{l_j h}}{\partial \phi_{h_j}} = A_{h_j} \cos(2\pi f_{h_j} t + \phi_{h_j})$

Die Verbesserungsgleichungen lauten dann für jede Beobachtung i

$$\begin{aligned}
 \hat{v} &= \frac{\partial f(\hat{x})}{\partial \hat{x}} \Delta \hat{x} - (b - f(x)) \\
 \Rightarrow \hat{v}^i &= \frac{\partial \Delta LOD_{l_j h}^i}{\partial a_0} \Big|_0 \Delta \hat{a}_0 + \frac{\partial \Delta LOD_{l_j h}^i}{\partial a_1} \Big|_0 \Delta \hat{a}_1 + \\
 &+ \frac{\partial \Delta LOD_{l_j h}^i}{\partial A_j} \Big|_0 \Delta \hat{A}_j + \frac{\partial \Delta LOD_{l_j h}^i}{\partial f_j} \Big|_0 \Delta \hat{f}_j + \frac{\partial \Delta LOD_{l_j h}^i}{\partial \phi_j} \Big|_0 \Delta \hat{\phi}_j + \\
 &+ \frac{\partial \Delta LOD_{l_j h}^i}{\partial A_{h_j}} \Big|_0 \Delta \hat{A}_{h_j} + \frac{\partial \Delta LOD_{l_j h}^i}{\partial f_{h_j}} \Big|_0 \Delta \hat{f}_{h_j} + \frac{\partial \Delta LOD_{l_j h}^i}{\partial \phi_{h_j}} \Big|_0 \Delta \hat{\phi}_{h_j} - \\
 &- (\Delta LOD_{l_j h}^i - \Delta LOD_{l_j h}^i \Big|_0), \tag{4.10}
 \end{aligned}$$

oder bei insgesamt n Beobachtungen in Matrix-Schreibweise $\hat{v} = A \Delta \hat{x} - w$. Hierbei sind

- $f(x)$ der funktionale Zusammenhang zwischen Beobachtungen und Unbekannten (vgl. Formel 4.9),
- $b = [\Delta LOD^1 \quad \dots \quad \Delta LOD^n]^T$ die Beobachtungen, d.h. die Zeitreihe selbst,
- $A = \begin{bmatrix} \frac{\partial \Delta LOD_{l_j h}^1}{\partial a_0} \Big|_0 & \dots & \frac{\partial \Delta LOD_{l_j h}^1}{\partial \phi_{h_j}} \Big|_0 \\ \vdots & & \vdots \\ \frac{\partial \Delta LOD_{l_j h}^n}{\partial a_0} \Big|_0 & \dots & \frac{\partial \Delta LOD_{l_j h}^n}{\partial \phi_{h_j}} \Big|_0 \end{bmatrix}$ die Jakobimatrix,
- $\Delta \hat{x} = [\Delta \hat{a}_0 \quad \Delta \hat{a}_1 \quad \Delta \hat{A}_j \quad \Delta \hat{f}_j \quad \Delta \hat{\phi}_j \quad \Delta \hat{A}_{h_j} \quad \Delta \hat{f}_{h_j} \quad \Delta \hat{\phi}_{h_j}]^T$ die Zuschläge an den Näherungswerten \hat{x} für die Unbekannten und
- $w = \begin{bmatrix} \Delta LOD_{l_j h}^1 - \Delta LOD_{l_j h}^1 \Big|_0 \\ \vdots \\ \Delta LOD_{l_j h}^n - \Delta LOD_{l_j h}^n \Big|_0 \end{bmatrix}$ der Widerspruchsvektor.

Die Schätzung der Unbekannten erfolgt nach den Formeln

$$\Delta \hat{x} = (A' P_{bb} A)^{-1} A' P_{bb} w \tag{4.11}$$

$$\hat{x} = \hat{x} + \Delta \hat{x}. \tag{4.12}$$

Die Verbesserungen und die ausgeglichenen Beobachtungen können wir dann mit

$$\hat{v} = A \Delta \hat{x} - w \tag{4.13}$$

$$\hat{b} = b + \hat{v} \tag{4.14}$$

berechnen (Ebner, 1998).

Erfüllen die Ausgleichungsergebnisse \hat{x} und \hat{b} die Beobachtungsgleichungen $\hat{b} = f(\hat{x})$ genügend genau, so ist die Ausgleichung abgeschlossen. Andernfalls müssen wir die ausgeglichenen Ergebnisse als neue Näherungswerte für die Unbekannten einführen und solange iterieren, bis die Bedingung erfüllt ist. Anschließend schätzen wir den Varianzfaktor und die Kovarianzmatrix der Beobachtungen:

$$\hat{\sigma}_0^2 = \frac{\hat{v}' P_{bb} \hat{v}}{r} \tag{4.15}$$

$$\hat{K}_{bb} = \hat{\sigma}_0^2 Q_{bb} \tag{4.16}$$

Die Redundanz r ist die Differenz der Anzahl an Beobachtungen und Unbekannten. Die Genauigkeiten der Unbekannten können wir abschließend durch folgende Formel berechnen:

$$\hat{K}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2 (A' P_{bb} A)^{-1} \tag{4.17}$$

Die Standardabweichungen der Unbekannten entsprechen den Quadratwurzeln der Hauptdiagonalelemente von $\hat{K}_{\hat{x}\hat{x}}$. Als initiale Näherungswerte für die Ausgleichung wählen wir

$$\hat{x} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \hat{A}_j \\ \hat{f}_j \\ \hat{\phi}_j \\ \hat{A}_{hj} \\ \hat{f}_{hj} \\ \hat{\phi}_{hj} \end{bmatrix} = \begin{bmatrix} 0 & ms \\ 0 & ms/Tag \\ 1 & ms \\ 1/365 & 1/Tag \\ 0 & rad \\ 1 & ms \\ 2/365 & 1/Tag \\ 0 & rad \end{bmatrix}. \quad (4.18)$$

Sollte die Ausgleichung nicht konvergieren, d.h. die Differenz zwischen ausgeglichenen Beobachtungen und der Funktion an Stelle der geschätzten Unbekannten wird von einer Iterationsstufe zur nächsten größer, so haben wir vermutlich die initialen Näherungswerte zu schlecht gewählt. Die oben angeführten Näherungswerte erwiesen sich für alle in dieser Arbeit behandelten Fälle als ausreichend.

Variationsanalyse

Um Aussagen über die zeitliche Variation der geschätzten Parameter machen zu können, wählen wir ein Zeitfenster konstanter Länge, das über den gesamten Datensatz verschoben wird. Im Falle von ΔLOD setzen wir zunächst ein Zeitfenster von 4 Jahren an, um die Schwebungsperiode (Stöcker, 1994) von $T_s = \frac{1}{f_{hj} - f_j} \approx 365$ Tagen ausreichend gut repräsentieren zu können. Der *Berechnungspunkt* liegt dabei immer genau in der Mitte des Zeitfensters. Da wir das Fenster von den jüngeren (genaueren) Daten hin zu den älteren (ungenaueren) Daten bewegen wollen, erstreckt sich das erste Zeitfenster vom 1.1.1995 bis zum 31.12.1998. Der Berechnungspunkt entspricht also dem 31.12.1996. Für diesen Zeitraum werden die oben beschriebenen 8 Parameter und deren Genauigkeiten geschätzt. Anschließend wird das Fenster auf der Zeitachse um 10 Tage in Richtung Vergangenheit verschoben und die Parameter erneut berechnet. Die geschätzten Unbekannten des Vorgängerfensters dienen dabei als neue Näherungswerte. Da wir die kurzperiodischen Schwankungen von ΔLOD der nächsten 10 Tage vorhersagen wollen, legen wir den Verschiebungsbetrag auf 10 Tage fest. Die Differenz der Parameter zwischen zwei aufeinanderfolgenden Berechnungspunkten ist also ein Maß für die Veränderung der bekannten Komponente von ΔLOD innerhalb von 10 Tagen.

Die Ergebnisse der Berechnung sind in den Abbildungen 4.5, 4.6 und 4.7 graphisch dargestellt. Die Standardabweichungen sind exemplarisch für 30 Berechnungspunkte durch rote Fehlerbalken veranschaulicht. In Abbildung 4.5 sind zusätzlich die mittlere Periodendauer eines Jahres von 365.25 Tagen und die eines halben Jahres von $\frac{365.25}{2}$ Tagen in grüner Farbe eingezeichnet. Deutlich ist hier die starke Variation der jährlichen Periode (ca. 353 bis 382 Tage) im Vergleich zur halbjährlichen (ca. 179 bis 186) zu erkennen. Auffällig ist auch die anscheinend sehr hohe Korreliertheit sowohl zwischen Perioden- und Phasenvariation (vgl. Abbildung 4.5) als auch zwischen Offset- und Steigungsvariation des linearen Anteils (vgl. Abbildung 55). Diesen Verdacht bestätigt uns Abbildung 4.7. Die Korrelation liegt bei ungefähr 99.995 Prozent. Diese zunächst ungewöhnlich erscheinende Tatsache ist nach kurzer Überlegung leicht nachzuvollziehen. Ausschlaggebend ist das in der Ausgleichung verwendete Zeitmaß. Da wir für die Zeitvariable t das *modifizierte julianische Datum* (MJD) verwenden, liegt der Nullpunkt auf der Zeitachse sehr weit vor unserem Berechnungszeitraum. Da $MJD = JD - 2400000.5$, ist der 17.11.1858 der Nullpunkt für t . Würden wir unseren Ursprung in die Mitte des Berechnungszeitraumes legen, so wären die Korrelationen in der Nähe des Ursprungs sehr klein. Dieses Phänomen ist anhand der Korrelation zwischen Offset und Steigung am besten erklärbar (siehe Abbildung 4.7, mittlere Graphik). Befinden wir uns in Ursprungsnähe, so wirkt sich eine Veränderung der Steigung einer Geraden nur gering auf den Achsenabschnitt (Offset) aus. Entfernen wir uns von ihm, so bewirkt die gleiche Steigungsänderung eine immer größere Änderung des Offsets. Dies verdeutlicht Abbildung 4.8. Erhöhen wir die Steigung in einem Punkt, so verringert sich der Wert des Achsenabschnitts. Damit ist das negative Vorzeichen der Korrelation zu erklären.

Analoges gilt für die Korrelation zwischen Phase und Frequenz (vgl. Abbildung 4.7 oben). Das Argument der Winkelfunktion $\sin(2\pi ft + \phi)$ können wir als Geradengleichung auffassen. Die Steigung entspricht dabei der Frequenz f , der Offset entspricht der Phase ϕ . Da hier jedoch nicht die Korrelation zwischen Phase und Frequenz, sondern zwischen Phase und Periode aufgetragen ist, hat der Korrelationskoeffizient ein positives Vorzeichen.

Deutlich wird in beiden Graphiken, dass mit zunehmendem Abstand vom Ursprung der Betrag der Korrelation ansteigt.

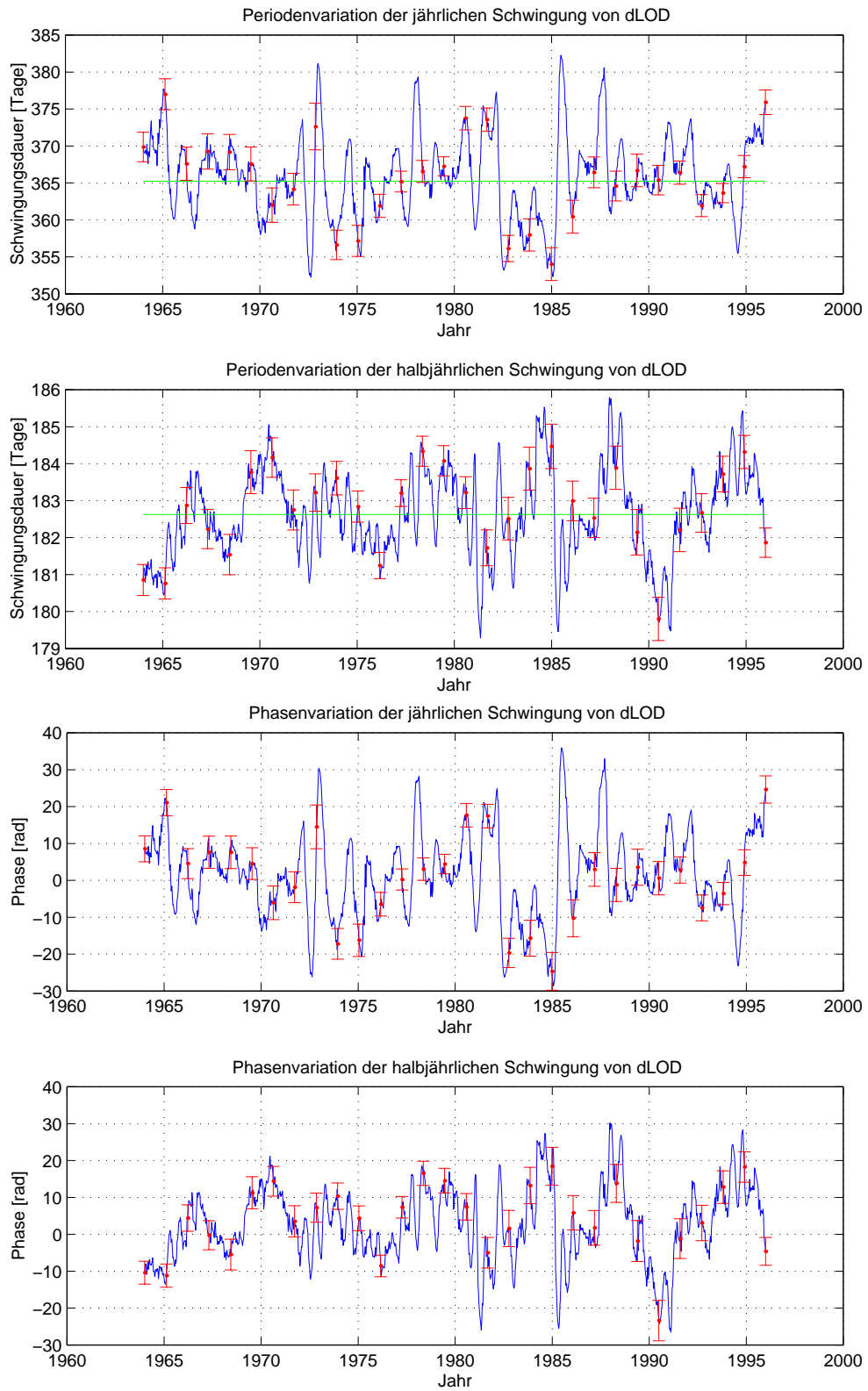


Abbildung 4.5: Perioden- und Phasenvariation von Δ LOD

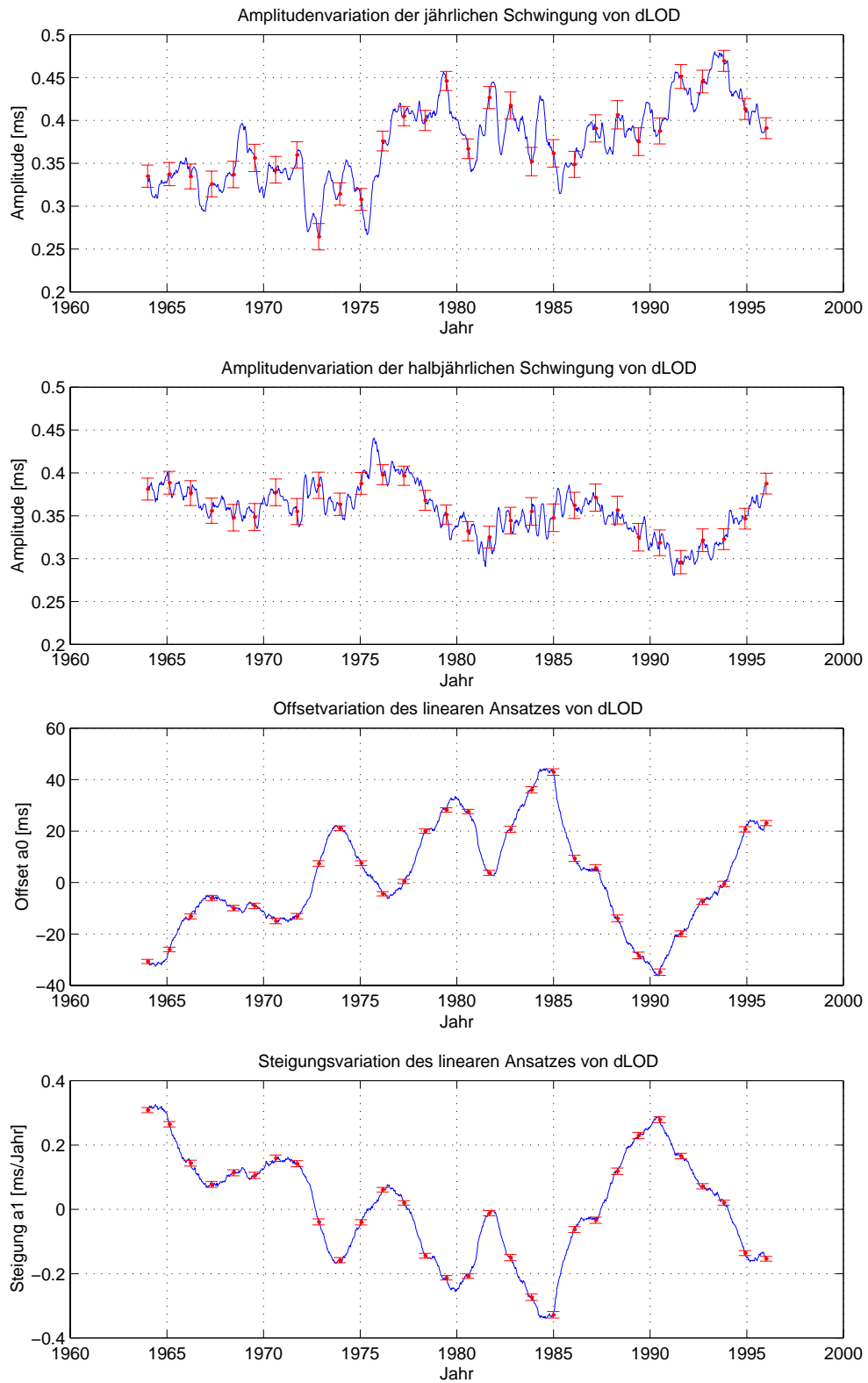
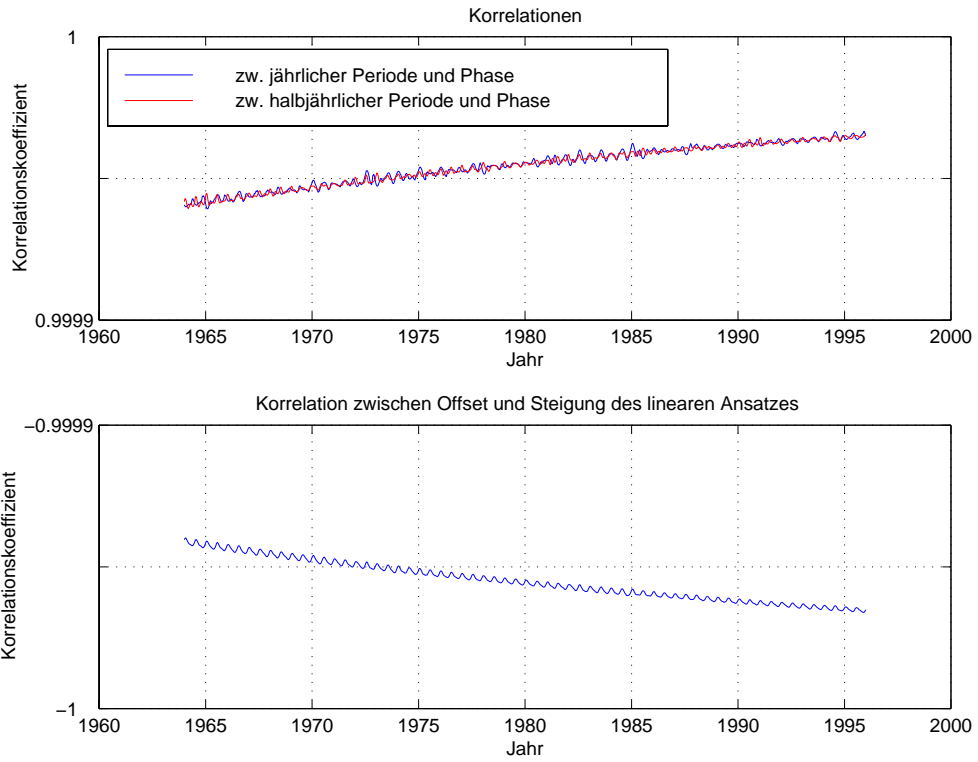
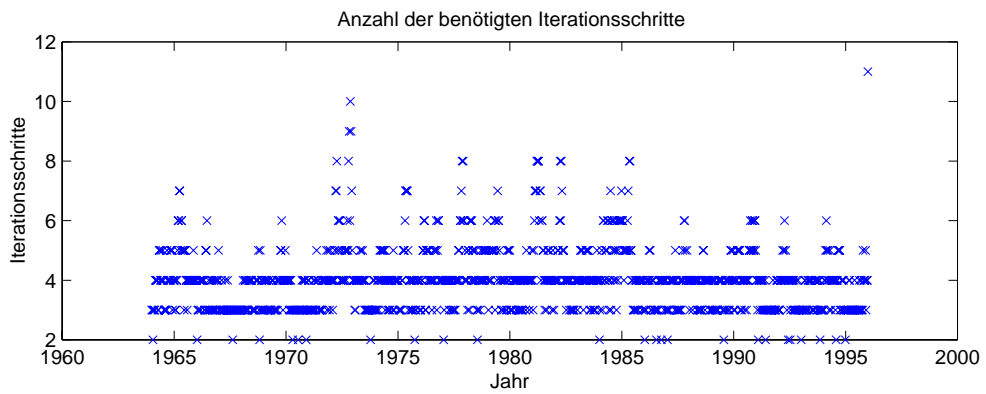


Abbildung 4.6: Variation der Amplitude und der Parameter des linearen Anteils von ΔLOD



(a) Korrelation



(b) Iterationsschritte

Abbildung 4.7: Korrelationen zwischen den geschätzten Parametern von Δ LOD und der Anzahl der benötigten Iterationsschritte während der Ausgleichung

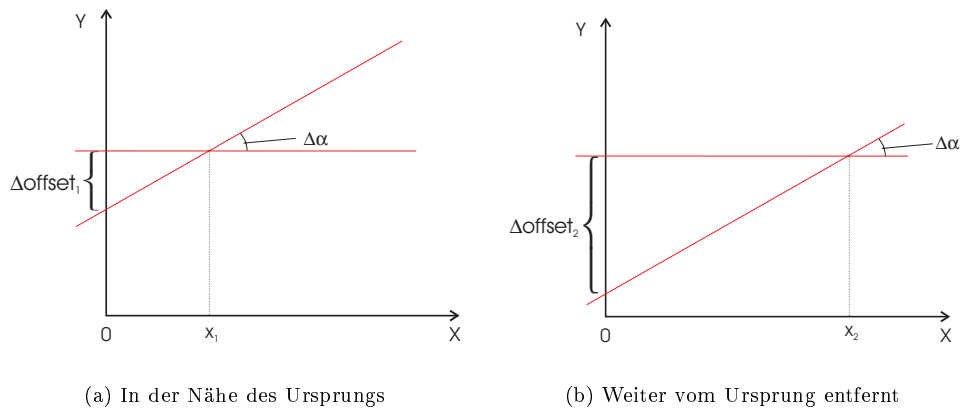


Abbildung 4.8: Korrelation zwischen Achsenabschnitt und Steigung

Die Obergrenze für die Anzahl an Iterationsschritten wurde auf 40 festgelegt. Diese Grenze wurde bei keinem Berechnungspunkt erreicht (vgl. Abbildung 4.7 unten). Mit Hilfe der benötigten Anzahl an Iterationsschritten während der Ausgleichung können wir auf die Stärke der Veränderung der Parameter von einem Berechnungszeitpunkt zum nächsten schließen. Da die Ergebnisse eines Berechnungszeitpunktes als neue Näherungswerte eingeführt werden, sind umso mehr Iterationsschritte nötig, je stärker die Parameter variieren.

Die folgende Tabelle zeigt die numerischen Ergebnisse der Analyse⁴:

ΔLOD		Über gesamten Zeitraum geschätzte Werte				Differenz		
Parameter	Einheit	μ	σ	Min.	Max.	μ _d	σ _d	Max.
a ₀	ms	1.458	19.511	-36.215	44.357	-0.046	0.587	2.368
a ₁	ms/Jahr	0.010	0.160	-0.339	0.326	0.001	0.005	0.019
A _j	ms	0.374	0.046	0.264	0.480	0.000	0.004	0.014
T _j = 1/f _j	Tage	365.846	5.489	352.236	382.265	-0.005	1.119	5.849
φ _j	rad	1.383	11.462	-28.601	35.957	-0.014	2.303	11.234
A _{h,j}	ms	0.356	0.028	0.281	0.441	0.000	0.003	0.014
T _{h,j} = 1/f _{h,j}	Tage	182.637	1.182	179.287	185.790	-0.001	0.281	1.535
φ _{h,j}	rad	2.763	10.047	-26.560	30.205	-0.005	2.376	13.010

Die letzten drei Spalten in der Tabelle beziehen sich auf den Unterschied der geschätzten Parameter zwischen zwei aufeinanderfolgenden Berechnungspunkten im Abstand von 10 Tagen. Mit Hilfe der Standardabweichungen dieser Differenzen können wir über das Fehlerfortpflanzungsgesetz die Genauigkeit einer Vorhersagemethode bestimmen, die nur auf Extrapolieren der Parameter beruhen würde:

$$\sigma_{\Delta LOD} = \sqrt{\frac{\partial \Delta LOD}{a_0} \sigma_{d_{a_0}}^2 + \frac{\partial \Delta LOD}{a_1} \sigma_{d_{a_1}}^2 + \dots + \frac{\partial \Delta LOD}{\phi_{hj}} \sigma_{d_{\phi_{hj}}}^2} = 1.82 \text{ ms} \tag{4.19}$$

Die Ableitungen wurden dabei jeweils an der Stelle der Mittelwerte der geschätzten Parameter (siehe obige Tabelle) berechnet. Die durch ein solches Verfahren erzielte Genauigkeit ist nicht zufriedenstellend. Subtrahieren wir im Vorfeld allerdings das Modell der zonalen Gezeiten mit Perioden von 5 Tagen bis 18.6 Jahren und berechnen anschließend erneut die Ausgleichungsergebnisse, so reduziert sich der Fehler auf $\sigma_{\Delta LOD} = 0.99 \text{ ms}$. Dies entspricht einer Genauigkeitssteigerung von 84 Prozent. Mit Hilfe der neuronalen Netze soll nun versucht werden, den Fehler dieses Modells so weit wie möglich zu verkleinern.

4.1.4 Reduktion von ΔLOD um alle bekannten Einflüsse

Die in den Kapiteln 4.1.2 und 4.1.3 modellierten und berechneten bekannten Einflüsse auf ΔLOD, können nun von der Datenreihe subtrahiert werden. Berücksichtigt werden sowohl zonale Gezeiten als auch Ozeangezeiten

⁴μ entspricht dem Mittelwert, σ der Standardabweichung

mit Perioden von 5 Tagen bis 18.6 Jahren, der lineare Anteil und die jährliche und halbjährliche Variation. Die reduzierte Datenreihe ΔLOD_{red} kann mathematisch durch

$$\Delta LOD_{red} = \Delta LOD - (\Delta LOD - \Delta LODS) - \Delta LOD_{lin,j,h,j}$$

ausgedrückt werden (vgl. Formel 4.9 und 4.5). Zu beachten ist die richtige Reihenfolge der Korrekturen. Zunächst müssen die Gezeiten modelliert und reduziert werden. Die korrigierten $\Delta LODS$ -Werte dienen anschließend als Beobachtungen für die Ausgleichung zur Schätzung der Parameter des linearen Anteils und der saisonalen Einflüsse. Durch Einhaltung dieser Abfolge vermeiden wir eine doppelte Subtraktion der jährlichen und halbjährlichen Gezeiteneinflüsse. Die Ergebnisse sind in den Abbildungen 4.9 und 4.10 zusammengestellt. Vor allem anhand des Amplitudenspektrums in Abbildung 4.10 ist die Reduktion sehr gut nachvollziehbar. In der untersten Graphik sind fast keine auffälligen Spitzen mehr erkennbar. Lediglich bei einer Periode von ca. 120 Tagen verbleibt eine Restschwingung. Diese kann der Atmosphäre zugeordnet werden und ist folglich in der Gezeitenmodellierung nicht enthalten.

Die Vorhersage kurzfristiger Schwankungen in Kapitel 5.1 basiert auf der Grundlage dieser reduzierten Daten ΔLOD_{red} .

4.2 Erdrotation ($\Delta UT1$)

4.2.1 Darstellung der Zeitreihe und Frequenzanalyse

Die ursprüngliche Datenreihe $UT1 - UTC$ muss zunächst um die Schaltsekunden korrigiert werden, um das stetige $\Delta UT1 = UT1 - TAI$ zu erhalten (siehe Abbildung 4.11). Da $\Delta UT1$ in erster Näherung linear abnimmt, ist das Spektrum überwiegend von der Frequenzdarstellung einer Geraden geprägt. Tragen wir die Fouriertransformierte einer Geraden als Funktion der Periode in einer Graphik mit äquidistanter Skalierung auf, so erhalten wir wiederum eine Gerade, die dem Spektrum in Abbildung 4.11 unten ähnelt. Um Aussagen über die spektrale Zusammensetzung von $\Delta UT1$ machen zu können, muss also im Vorfeld der lineare Anteil abgezogen werden. Diese Reduktion wird in Kapitel 4.2.4 vorgenommen.

4.2.2 Gezeitenmodellierung

Analog der Gezeitenmodellierung für ΔLOD in Kapitel 4.1 werden auch hier die zonalen Gezeiten der festen Erde und die Meeresgezeiten mit Perioden von 5.6 Tagen bis 18.6 Jahren berücksichtigt. Als Ergebnis erhalten wir schließlich die Korrektur $UT1 - UT1S$ bzw. $\Delta UT1 - \Delta UT1S$, die später von der Datenreihe $\Delta UT1$ subtrahiert werden kann.

4.2.3 Saisonale Einflüsse und linearer Anteil

Wenden wir auf $\Delta UT1$ das gleiche Funktionalmodell an wie auf ΔLOD , so stellen wir fest, dass die Ausgleichung sehr instabil verläuft, da das Normalgleichungssystem in Bezug auf die Rechengenauigkeit singular wird. Die Folge ist, dass der Fehler von einer Iteration zur nächsten größer wird und die Ausgleichung somit nicht konvergieren kann. Die Ursache sind wohl die nach Reduktion des linearen Anteils immer noch sehr stark ausgeprägten langperiodischen und aperiodischen Variationen, die, vergleichbar mit einer Geraden, Störfrequenzen erzeugen. Diese Frequenzen dominieren und verschlucken die jährliche und halbjährliche Periode, sodass diese nur sehr schwer extrahiert werden können (Schuh, 1981).

Ein möglicher Lösungsweg für dieses Problem könnte folgendermaßen lauten: Halten wir in der Ausgleichung die Dauer eines Jahres P_y auf 365.25 Tagen fest, so können wir $\Delta UT1$ als Funktion der unbekannt Parameter a_0, a_1, b, c, d und e schreiben:

$$\Delta UT1_{lin,j,h,j} = a_0 + a_1 t + b \sin\left(\frac{2\pi}{P_y} t\right) + c \cos\left(\frac{2\pi}{P_y} t\right) + d \sin\left(\frac{4\pi}{P_y} t\right) + e \cos\left(\frac{4\pi}{P_y} t\right) \quad (4.20)$$

Dadurch, dass wir zusätzlich die Phase ϕ durch die Cosinus-Terme ausdrücken, ist diese Gleichung nun linear in den Unbekannten. Somit werden keine Näherungswerte für die Ausgleichung benötigt, und die Schätzung der Unbekannten erfolgt in nur einem Schritt. Die nun konstanten Frequenzen können wir in Kauf nehmen, da die jährliche und halbjährliche Periode nur geringfügig von den angenommenen Werten abweichen.

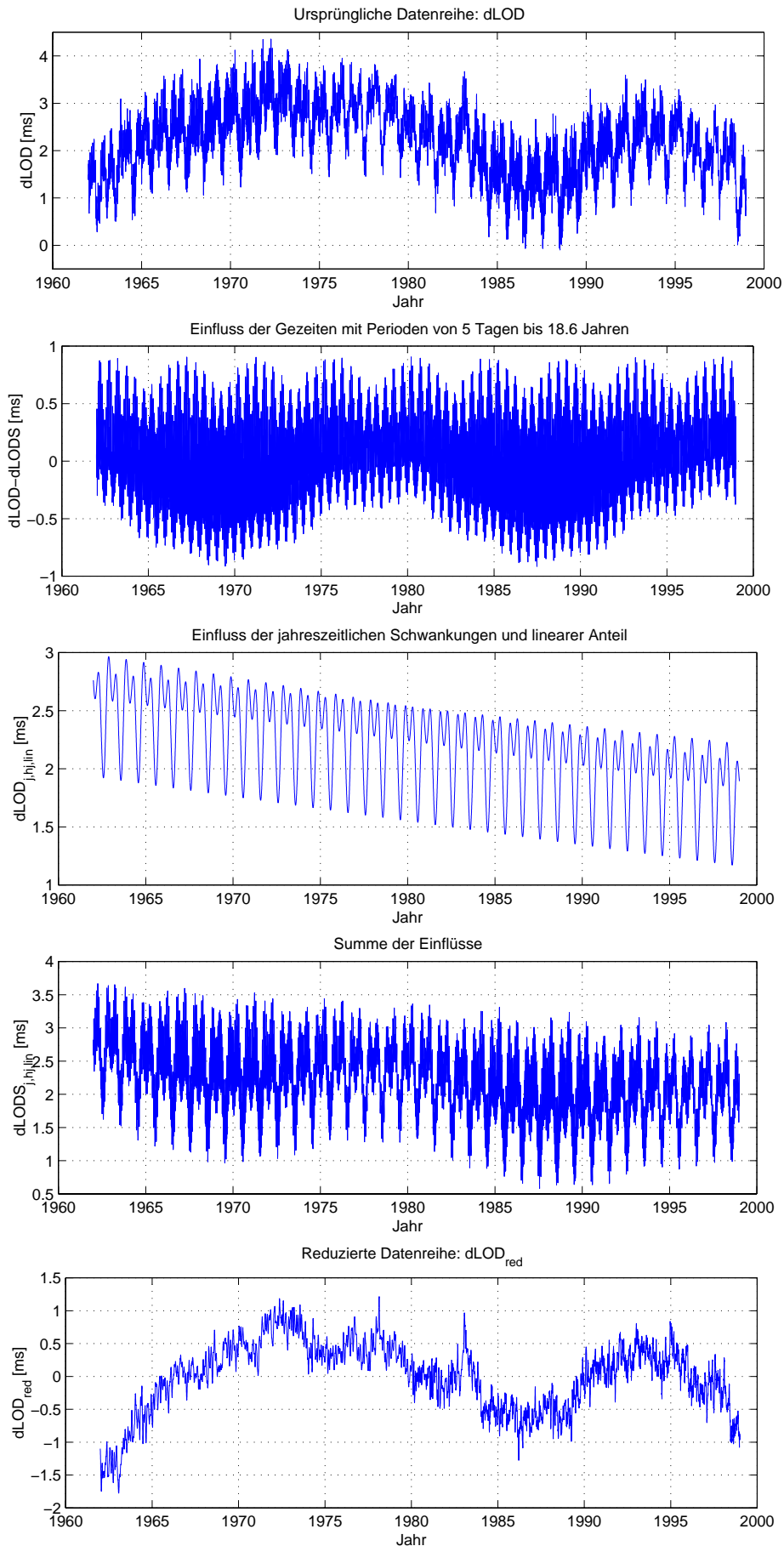


Abbildung 4.9: Reduktion der Datenreihe ΔLOD im Zeitbereich

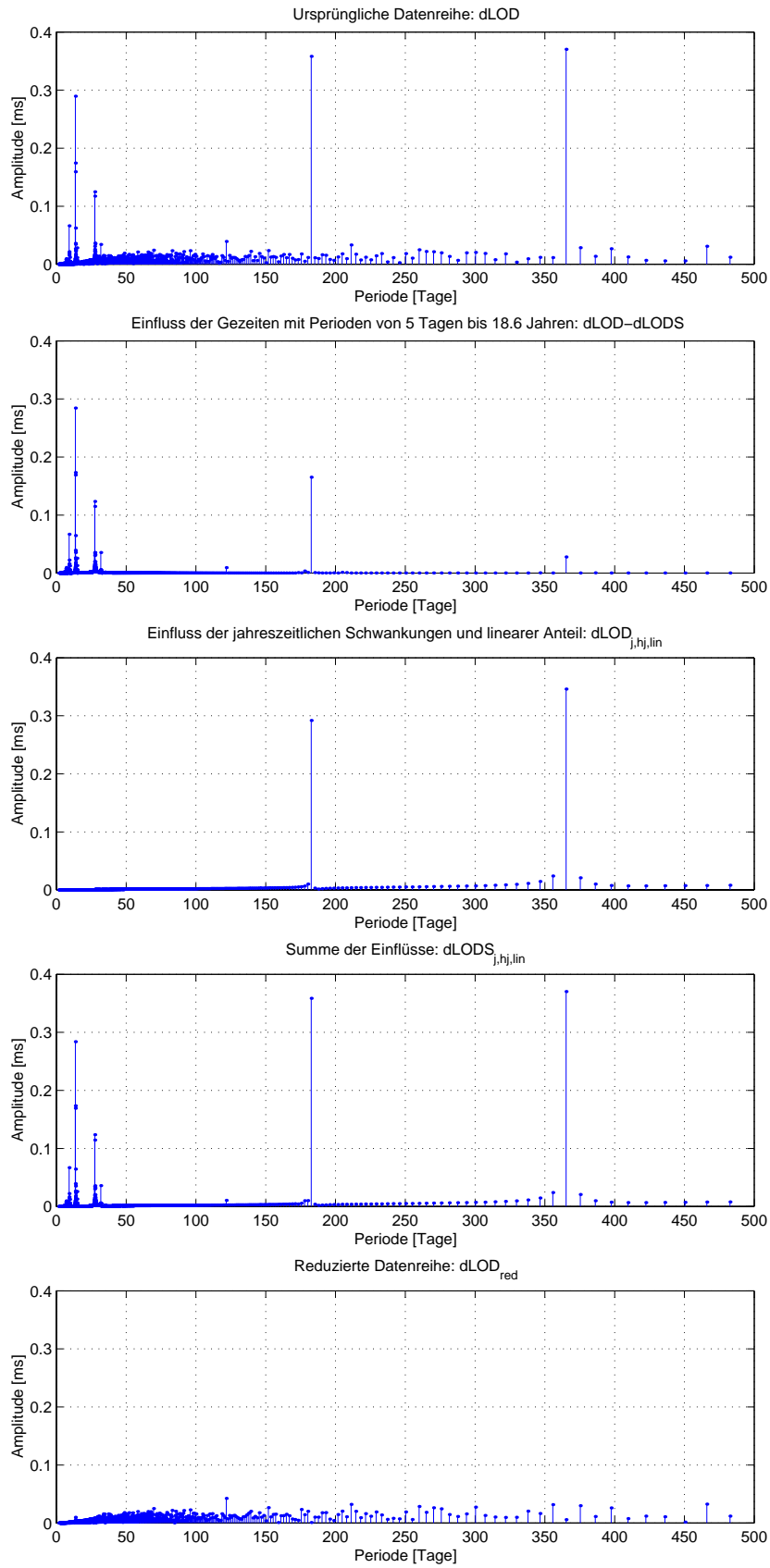


Abbildung 4.10: Reduktion der Datenreihe ΔLOD im Frequenzbereich

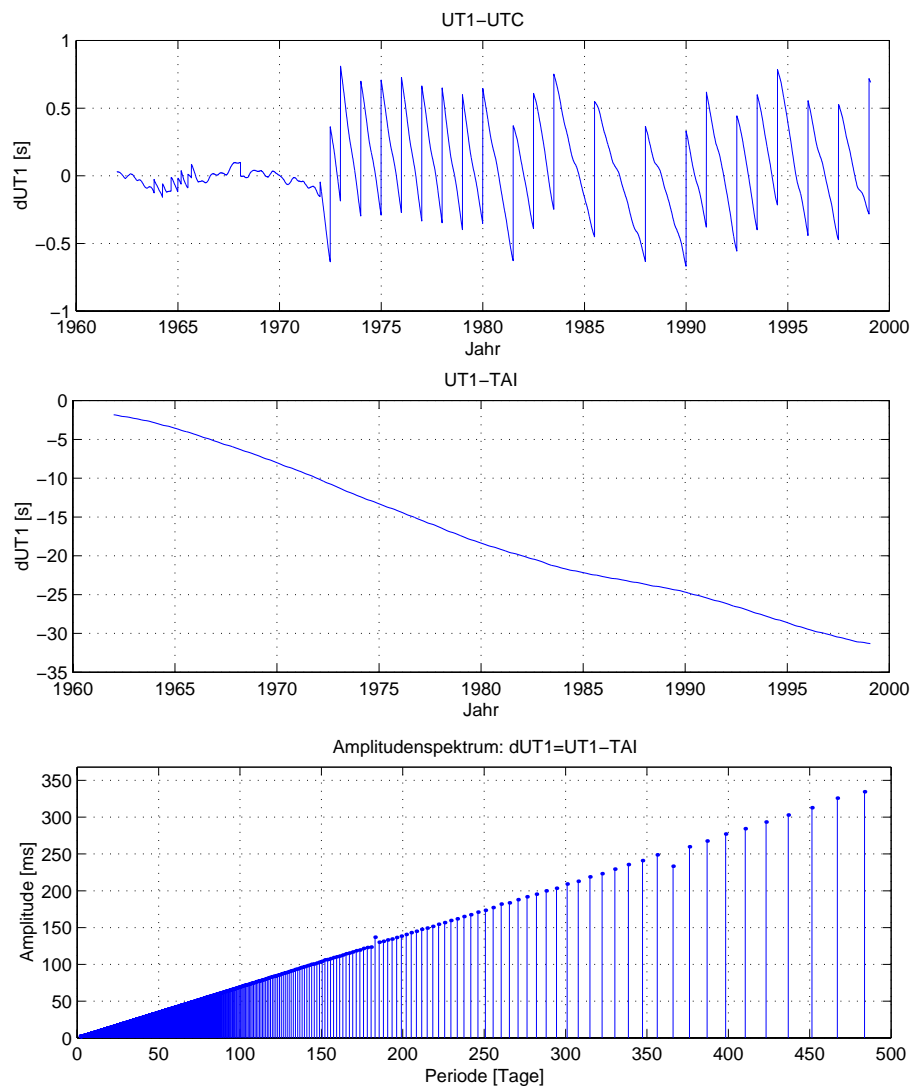


Abbildung 4.11: $\Delta UT1$ und Spektrum

Für die Variationsanalyse wählen wir ebenfalls ein Zeitfenster von 4 Jahren und schätzen für dessen Mittelpunkt - den Berechnungszeitpunkt - die unbekannt Parameter. Nach der Ausgleichung können die Koeffizienten b und c bzw. d und e in Amplitude und Phase umgerechnet werden. Mit Hilfe der partiellen Ableitungen werden anschließend deren Standardabweichungen über das Fehlerfortpflanzungsgesetz ermittelt.

Die Abbildungen 4.12 und 4.13 zeigen das Ergebnis der Variationsanalyse. Die Korrelationen zwischen Achsenabschnitt und Steigung des linearen Anteils verlaufen ähnlich wie bei ΔLOD , daher wird von einer graphischen Darstellung abgesehen. Anhand der großen Standardabweichungen der jährlichen und halbjährlichen Amplitude können wir erkennen, dass trotz Linearisierung und festgehaltener Frequenzen nur eine sehr ungenaue Bestimmung möglich scheint. Folgende Übersicht zeigt die exakten numerischen Ergebnisse:

$\Delta UT1$		Über gesamten Zeitraum geschätzte Werte				Differenz		
Parameter	Einheit	μ	σ	Min.	Max.	μ_d	σ_d	Max.
a_0	s	82.424	20.703	45.017	117.3755	0.006	0.191	0.423
a_1	s/Jahr	-0.829	0.171	-1.128	-0.532	0.000	0.002	0.004
A_j	ms	20.661	4.231	5.095	30.916	-0.009	0.616	1.932
ϕ_j	rad	1.833	0.199	1.092	2.519	0.001	0.036	0.344
A_{hj}	ms	8.484	1.702	3.329	13.969	0.002	0.526	1.907
ϕ_{hj}	rad	-2.569	0.199	-3.243	-1.972	-0.000	0.039	0.446

Durch die sehr starken Variationen der Parameter des linearen Ansatzes a_0 und a_1 , die von den nicht-modellierbaren langperiodischen Restschwingungen hervorgerufen werden, ergibt sich mit Hilfe des Fehlerfortpflanzungsgesetzes für den 10. Vorhersagetag eine auf Extrapolation beruhende Prädiktionsgenauigkeit von $\sigma_{\Delta UT1} = 65.3$ ms. Dieser in keinsten Weise akzeptable Wert zeigt, dass die Vorhersage von UT1 sehr hohe Ansprüche an das Prädiktionsverfahren stellt.

4.2.4 Reduktion von $\Delta UT1$ um alle bekannten Einflüsse

Die Gleichung zur Reduktion der Datenreihe lautet:

$$\Delta UT1_{red} = \Delta UT1 - (\Delta UT1 - \Delta UT1_S) - \Delta UT1_{lin,j,hj} \tag{4.21}$$

Die Ergebnisse sind in den Abbildungen 4.14 und 4.15 zusammengefasst. In der Frequenzdarstellung der um den linearen Anteil reduzierten Datenreihe UT1-TAI (vgl. Abbildung 4.15, obere Graphik) sind die saisonalen Schwingungen nun wesentlich besser zu erkennen als in Abbildung 4.11. Sogar die hochfrequenten Gezeiteneinflüsse unterhalb von 30 Tagen sind jetzt deutlich wahrnehmbar. Die Amplituden all dieser Frequenzen sind bei der reduzierten Datenreihe nicht mehr vorhanden. Der Einfluss der Atmosphäre mit einer Periode von ca. 120 Tagen ist auch hier noch zu bemerken, wenn auch nicht so deutlich wie bei ΔLOD .

4.3 Polbewegung

4.3.1 Darstellung der Zeitreihe und Frequenzanalyse

Die Abbildungen 4.16 und 4.17 zeigen die x- und y-Komponente der Polbewegung jeweils im Zeit- und Ortsbereich. Es fällt auf, dass beide Komponenten einen ähnlichen Verlauf und ein fast identisches Spektrum aufweisen. Die y-Komponente zeigt einen etwas stärkeren linearen Trend und ist größer als die x-Komponente.

Vergleichen wir die Spektren mit der Frequenzdarstellung von ΔLOD (siehe Abbildung 4.1), so bestätigt sich, dass die Gezeiten der festen Erde keinen Einfluss auf die Polbewegung nehmen. Auch eine halbjährliche Schwingung ist aus dem Spektrum nur noch zu erahnen. Neben der jährlichen Schwingung ist dafür nun ein neuer Peak bei einer Periode von ca. 435 Tagen zu sehen. Dabei handelt es sich um die in Kapitel 2.1.2 beschriebene Chandlersche Periode.

Die hochfrequenten Meeresgezeiten mit eintägigen und halbtägigen Perioden, die sehr wohl Einfluss auf die Polbewegung haben, sind aus den in Kapitel 4.1.2 genannten Gründen nicht enthalten und dürfen daher auch nicht reduziert werden. Der Einfluss der Meeresgezeiten auf die Polbewegung mit Perioden von über einem Tag ist noch

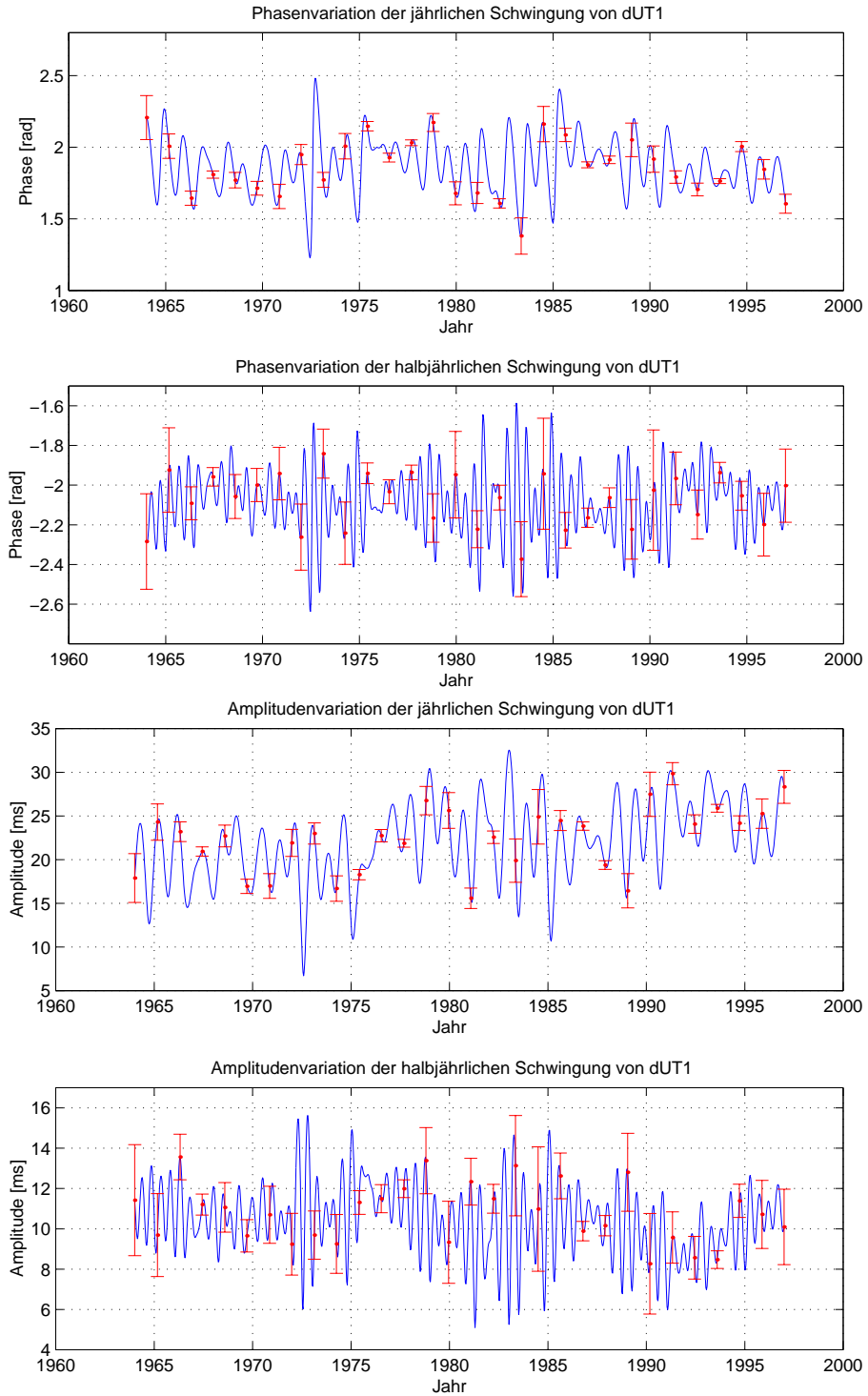


Abbildung 4.12: Variation der Phase und der Amplitude von $\Delta UT1$

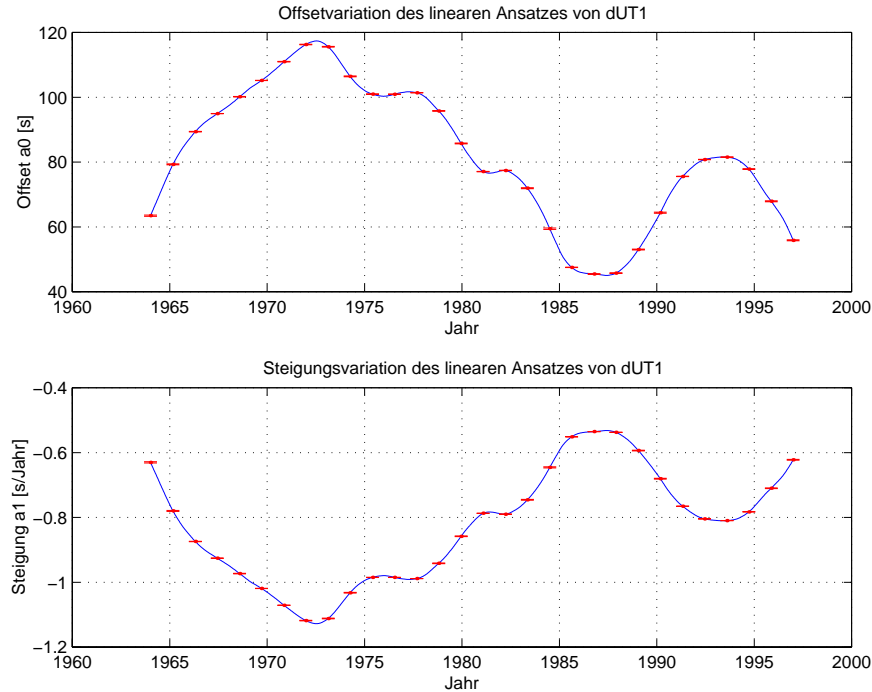


Abbildung 4.13: Variation der Parameter des linearen Anteils von $\Delta UT1$

geringer als ihr Einfluss auf die Rotationsgeschwindigkeit der Erde. Die Summe aller Partialzeiten ergibt lediglich einen Wert von ca. 0.1 mas (Seiler und Wunsch, 1995). Daher können wir auf eine Berücksichtigung dieses Effektes verzichten.

Auch um den Einfluss mit halbjährlicher Periode werden die Polbewegungsreihen nicht reduziert. Zum einen ist dessen Amplitude relativ klein und zum anderen variieren Amplitude und Frequenz so stark, dass die halbjährliche Schwingung für die Vorhersage keinesfalls als bekannt und damit a priori modellierbar angenommen werden darf.

4.3.2 Saisonale Einflüsse, Chandler-Schwingung und linearer Anteil

Mit Hilfe einer vermittelnden Ausgleichung werden die Parameter der Gleichungen

$$x_{lin,j,c} = a_0^x + a_1^x t + A_j^x \sin(2\pi f_j^x t + \phi_j^x) + A_c^x \sin(2\pi f_c^x t + \phi_c^x) \tag{4.22}$$

und

$$y_{lin,j,c} = a_0^y + a_1^y t + A_j^y \sin(2\pi f_j^y t + \phi_j^y) + A_c^y \sin(2\pi f_c^y t + \phi_c^y) \tag{4.23}$$

mit A_c, f_c, ϕ_c : Parameter der Chandlerschen Schwingung
 x : x – Komponente der Polbewegung
 y : y – Komponente der Polbewegung

analog zur Vorgehensweise aus Kapitel 4.1.3 geschätzt (Chao, 1984).

Aufgrund der Schwebungsfrequenz von ca. 6.2 Jahren legen wir die Breite unseres Zeitfensters auf 12 Jahre fest. Somit erstreckt sich das Intervall für die Berechnungspunkte vom 1.1.1968 bis zum 31.12.1992. Das Ergebnis der Variationsanalyse zeigen die Abbildungen 4.18 und 4.19. Da die Korrelationen ähnlich verlaufen wie bei der Variationsanalyse von ΔLOD , wurde auf ihre graphische Darstellung verzichtet.

Die numerischen Ergebnisse sind in den nachstehenden Tabellen zusammengefasst:

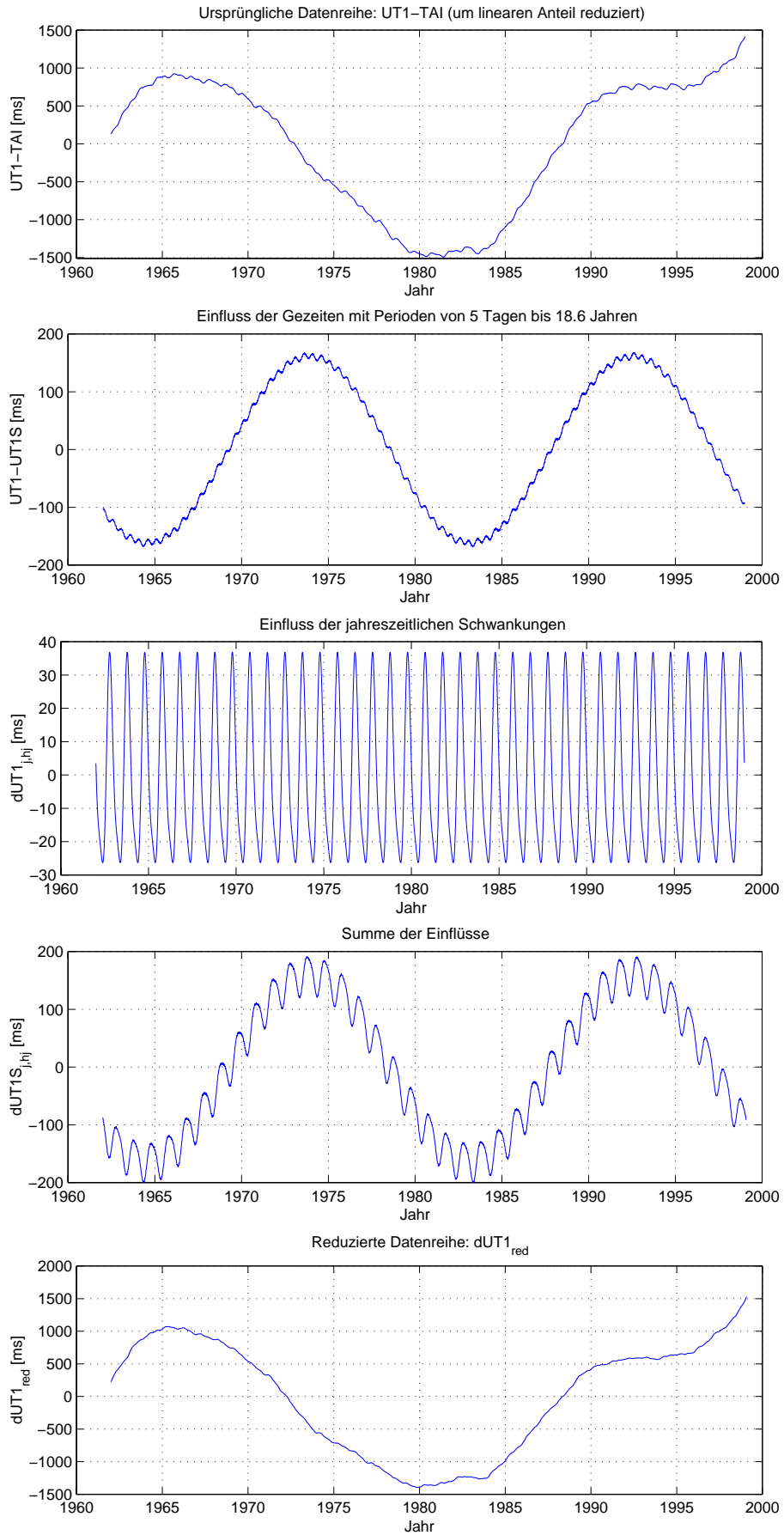


Abbildung 4.14: Reduktion der Datenreihe $\Delta UT1$ im Zeitbereich

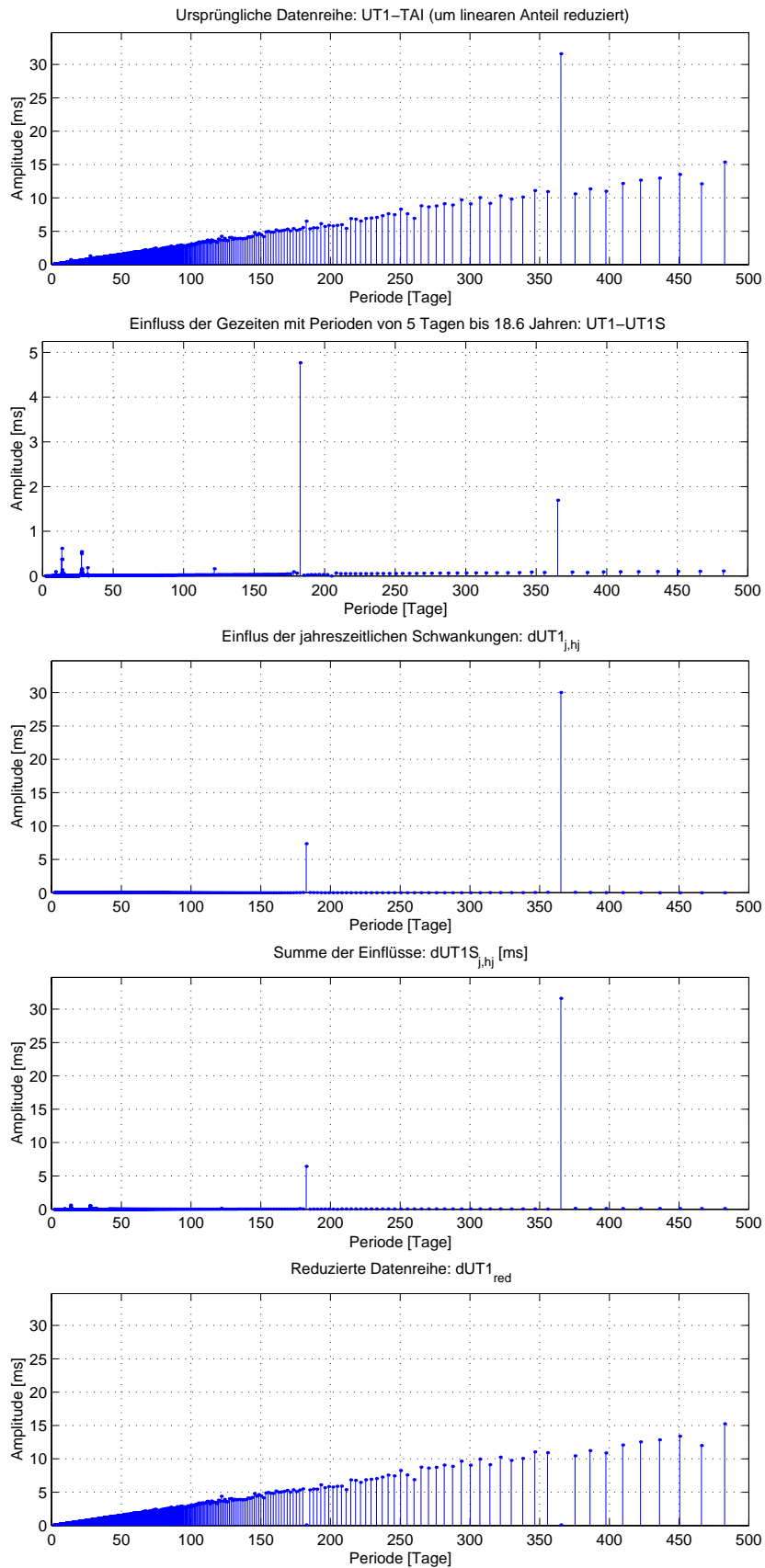


Abbildung 4.15: Reduktion der Datenreihe $\Delta UT1$ im Frequenzbereich

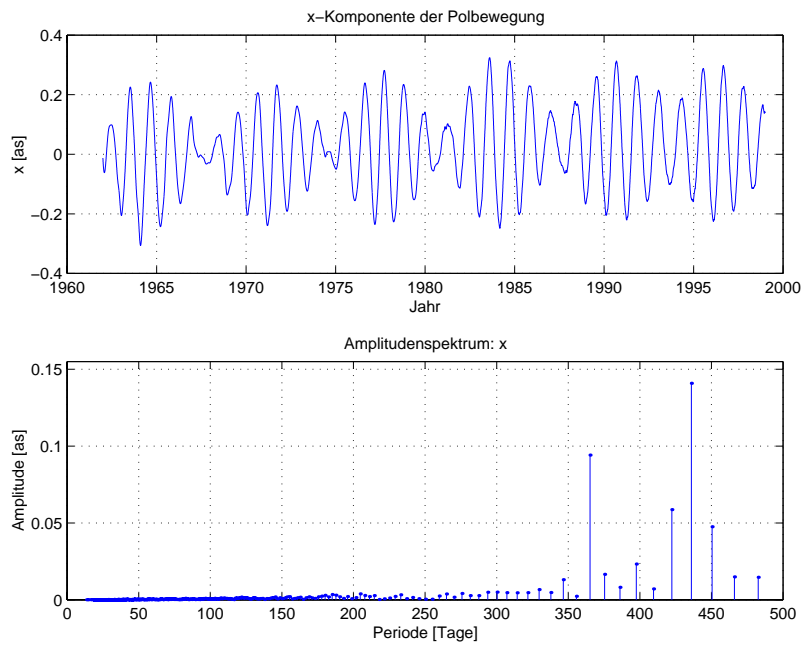


Abbildung 4.16: x-Komponente der Polbewegung und Spektrum

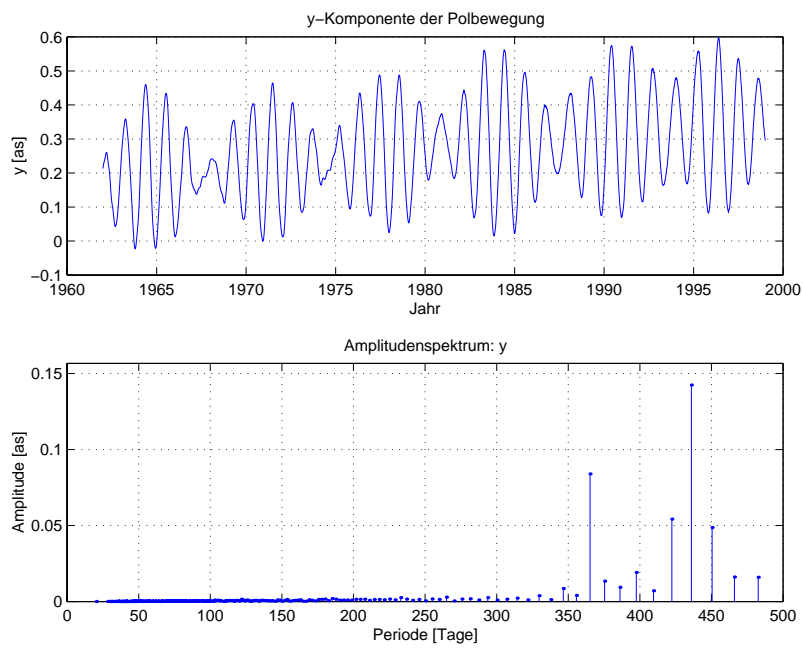


Abbildung 4.17: y-Komponente der Polbewegung und Spektrum

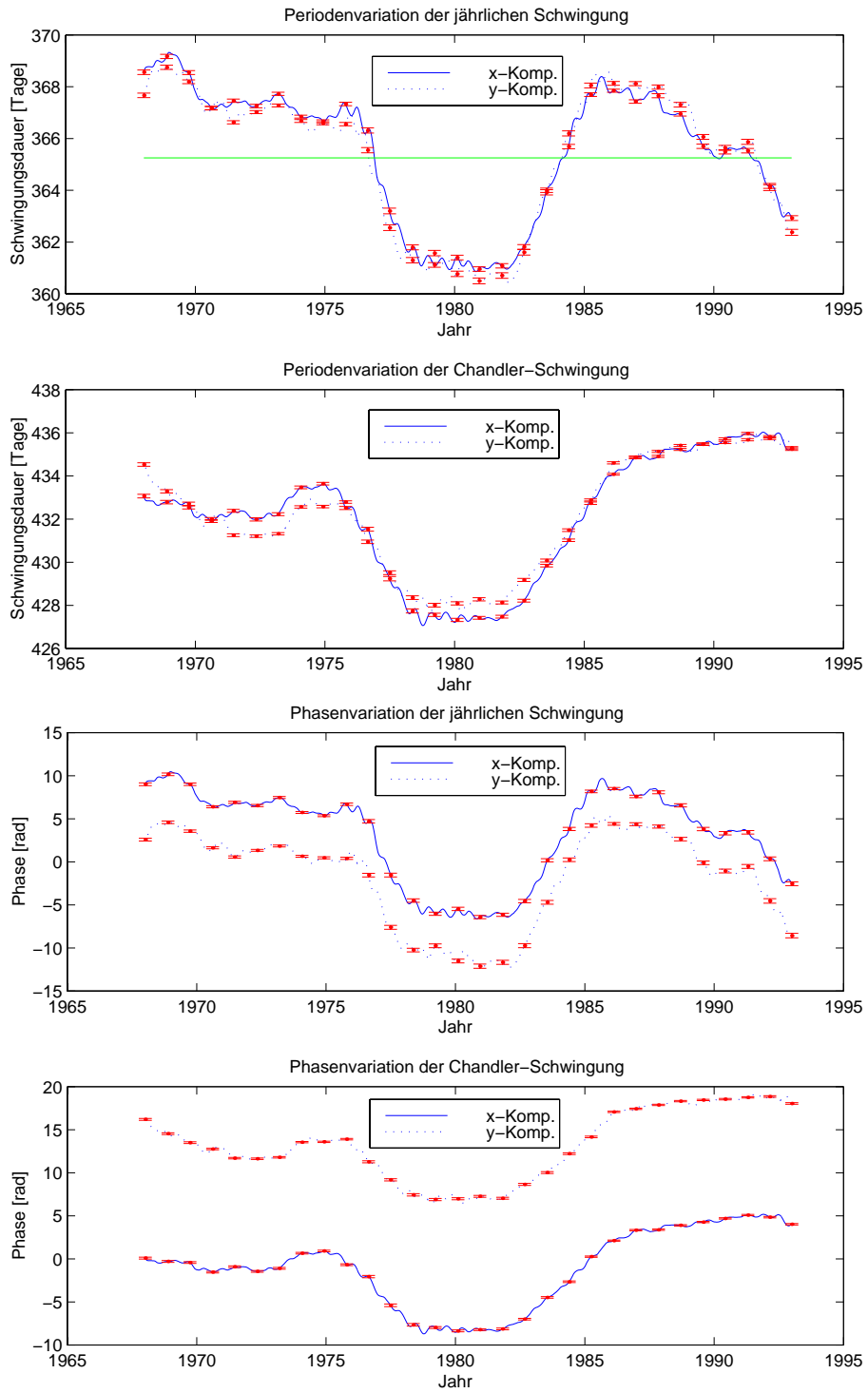


Abbildung 4.18: Perioden- und Phasenvariation der Polbewegung

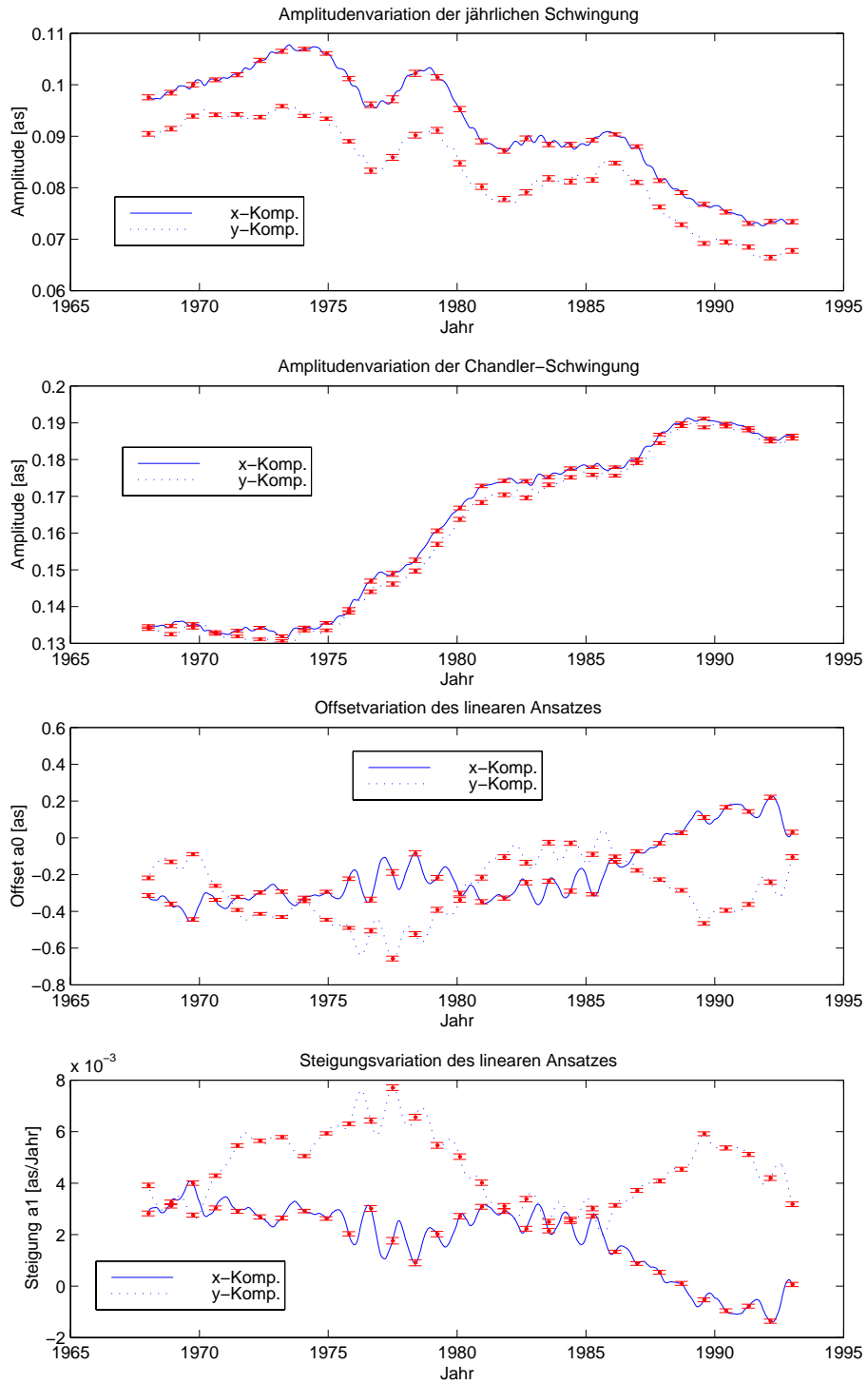


Abbildung 4.19: Variation der Amplitude und der Parameter des linearen Anteils der Polbewegung

x-Komponente		Über gesamten Zeitraum geschätzte Werte				Differenz		
Parameter	Einheit	μ	σ	Min.	Max.	μ_d	σ_d	Max.
a_0	as	-0.189	0.175	-0.453	0.225	-0.0001	0.006	0.018
a_1	$\mu\text{as}/\text{Jahr}$	4.995	3.803	-3.821	11.170	0.008	0.138	0.431
A_j	mas	92.322	10.112	73.127	108.548	0.026	0.139	0.446
$T_j = 1/f_j$	Tage	365.491	2.552	360.860	369.330	0.006	0.052	0.248
ϕ_j	rad	3.117	5.295	-6.623	10.471	0.013	0.109	0.497
A_c	mas	162.027	22.331	131.921	191.623	-0.057	0.138	0.455
$T_c = 1/f_c$	Tage	432.063	2.815	427.064	436.029	-0.002	0.041	0.151
ϕ_c	rad	-1.145	4.345	-8.709	5.184	-0.004	0.061	0.217

y-Komponente		Über gesamten Zeitraum geschätzte Werte				Differenz		
Parameter	Einheit	μ	σ	Min.	Max.	μ_d	σ_d	Max.
a_0	as	-0.291	0.166	-0.658	0.041	-0.0001	0.006	0.018
a_1	$\mu\text{as}/\text{Jahr}$	12.625	3.766	5.486	21.118	0.002	0.137	0.362
A_j	mas	83.618	8.954	66.328	95.873	0.025	0.143	0.444
$T_j = 1/f_j$	Tage	365.379	2.361	360.412	368.768	0.006	0.056	0.215
ϕ_j	rad	-1.821	5.503	-12.305	5.362	0.012	0.117	0.426
A_c	mas	159.691	21.922	130.162	190.241	-0.057	0.139	0.447
$T_c = 1/f_c$	Tage	432.133	2.575	427.731	435.932	-0.001	0.039	0.114
ϕ_c	rad	0.085	0.030	0.052	0.155	0.0001	0.001	0.003

Mit Hilfe des Fehlerfortpflanzungsgesetzes ergibt sich somit für den 10. Vorhersagetag eine auf Extrapolation beruhende Prädiktionsgenauigkeit von

$$\begin{aligned} \sigma_x &= 18.6 \text{ mas und} \\ \sigma_y &= 13.8 \text{ mas.} \end{aligned} \tag{4.24}$$

4.3.3 Reduktion der Polbewegung um alle bekannten Einflüsse

Die um den linearen Anteil, die jährliche Schwingung und die Chandler-Schwingung reduzierten Datenreihen x_{red} und y_{red} der Polbewegungskomponenten x und y erhalten wir aus

$$\begin{aligned} x_{red} &= x - x_{lin,j,c} \\ y_{red} &= y - y_{lin,j,c} \end{aligned} \tag{4.25}$$

Diese reduzierten Reihen sind wiederum Grundlage für die Vorhersage. In den Abbildungen 4.20 und 4.21 kann der Reduktionsvorgang graphisch nachvollzogen werden.

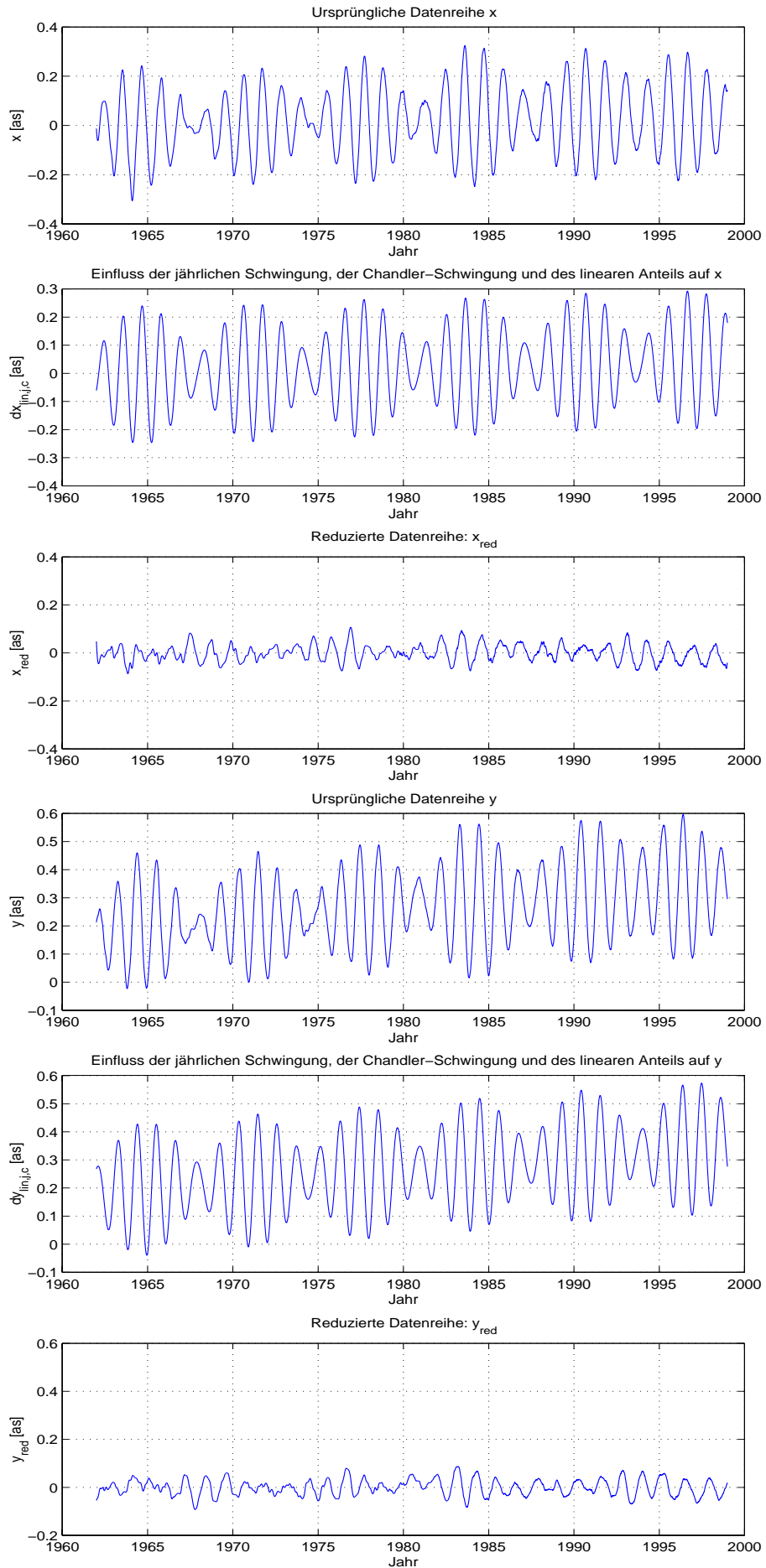


Abbildung 4.20: Reduktion der Polbewegung im Zeitbereich

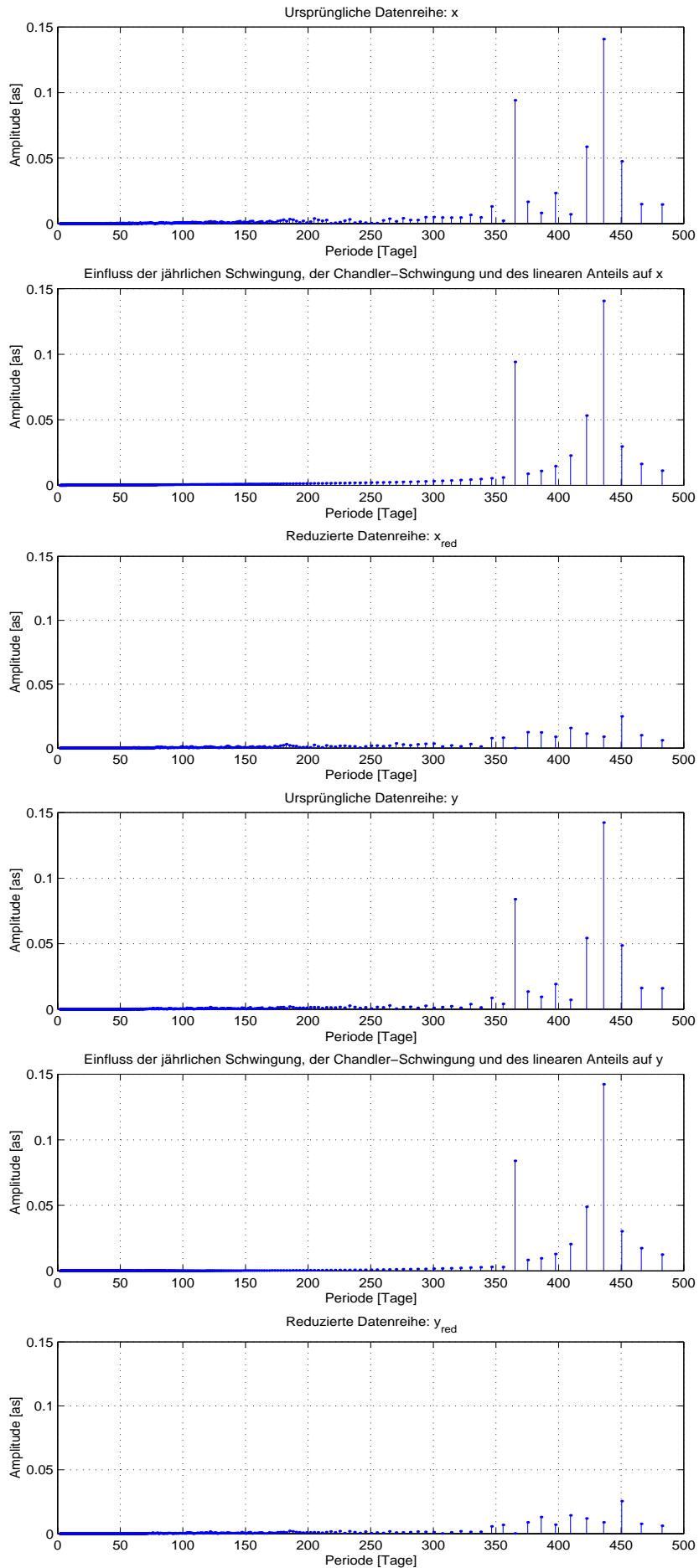


Abbildung 4.21: Reduktion der Polbewegung im Frequenzbereich

Kapitel 5

Vorhersage der Erdrotationsparameter

Wir unterscheiden im Folgenden zwischen kurzfristigen Schwankungen von bis zu 40 Tagen, mittelfristigen Variationen von bis zu einem Jahr und einer langfristigen Entwicklung mit Vorhersagen bis zu 40 Jahren.

Zunächst werden Prädiktionen im kurz- und mittelfristigen Bereich für die Erdrotationsparameter (Polbewegung, $\Delta UT1$ und ΔLOD) auf der Datengrundlage der C04-Reihe des IERS vorgenommen. Die gewonnenen Erkenntnisse werden dann auf die SPACE-Reihe von Gross (1999) angewendet und die Resultate miteinander verglichen.

Abschließend wird auf Grundlage der Vondrák-Reihe, die die Polbewegung ab dem Jahr 1900 beinhaltet, der Verlauf der x- und y-Komponente in 40 Jahren vorhergesagt. Bei der langfristigen Vorhersage beschränken wir uns deshalb auf die Polbewegung, weil zum einen gerade die starke Variation der Chandlerschen Schwingung eine Prädiktion sehr interessant macht und zum anderen genaue Werte für $\Delta UT1$ und ΔLOD erst ab dem Jahr 1956 gegeben sind und damit keine langfristige Vorhersage zulassen.

5.1 Kurzfristige Schwankungen

In diesem Kapitel werden die Erdrotationsparameter auf der Grundlage der in Kapitel 4 reduzierten Datenreihe C04 mit neuronalen Netzen vorhergesagt. Da die Genauigkeiten der C04-Parameter des IERS erst ab 1980 an die heute üblichen Vorhersagegenauigkeiten der kurzfristigen Schwankungen heranreicht, wird lediglich der Zeitraum vom 1.1.1980 bis zum 31.12.1998 als Datengrundlage herangezogen.

5.1.1 Vorhersage der Polbewegung

Die x-Komponente der Polbewegung wird in den ersten Versuchen dazu verwendet werden, eine optimale Netztopologie für ein feedforward-Netz zu finden. Gleichzeitig werden die Auswirkungen verschiedener Mustereinteilungen analysiert. Die y-Komponente wird dann mit diesen optimalen Einstellungen vorhergesagt.

Als Vergleichskriterium für die Qualität unterschiedlicher Konfigurationen wird jeweils der RMS-Fehler des 10. Vorhersagetages herangezogen.

Abbildung 5.1 zeigt für den oben genannten Zeitraum von 1980 bis 1998 die dreidimensionale Darstellung der Polbewegung, des geschätzten Modells und der um das Modell reduzierten Datenreihe.

Konfiguration 1

Zunächst werden die Daten von 1980 bis 1983 ausgespart, da 1984 nochmals eine Genauigkeitssteigerung erzielt werden konnte.

In einem ersten Versuch wird ein Netz mit fünf Ein- und Ausgabeneuronen getestet. Sowohl das Eingabe- als auch das Ausgabemuster werden mit [1 2 3 5 10] festgelegt (zur Mustereinteilung siehe Kapitel 3.2.1). Verschiedene Netztopologien mit unterschiedlicher Anzahl an versteckten Schichten und Neuronen pro Schicht werden einander gegenübergestellt. Die chronologisch letzten 10 Prozent der Muster werden zum Validieren genutzt. Von den

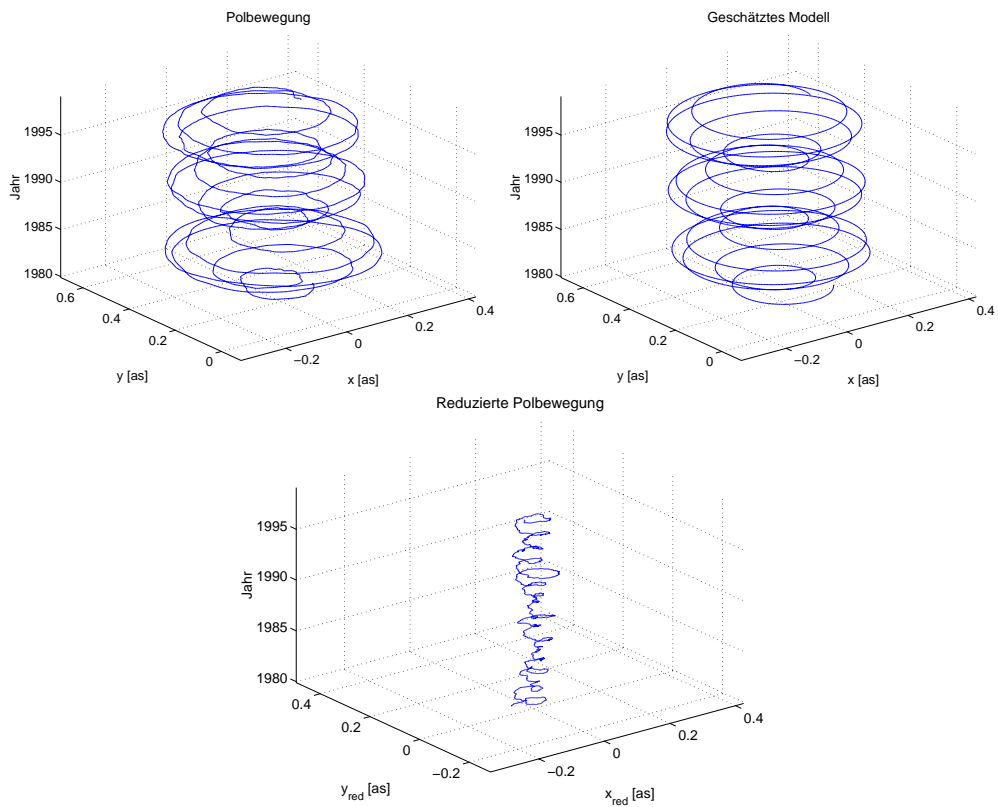


Abbildung 5.1: Polbewegung, Ausgleichende und Residuen

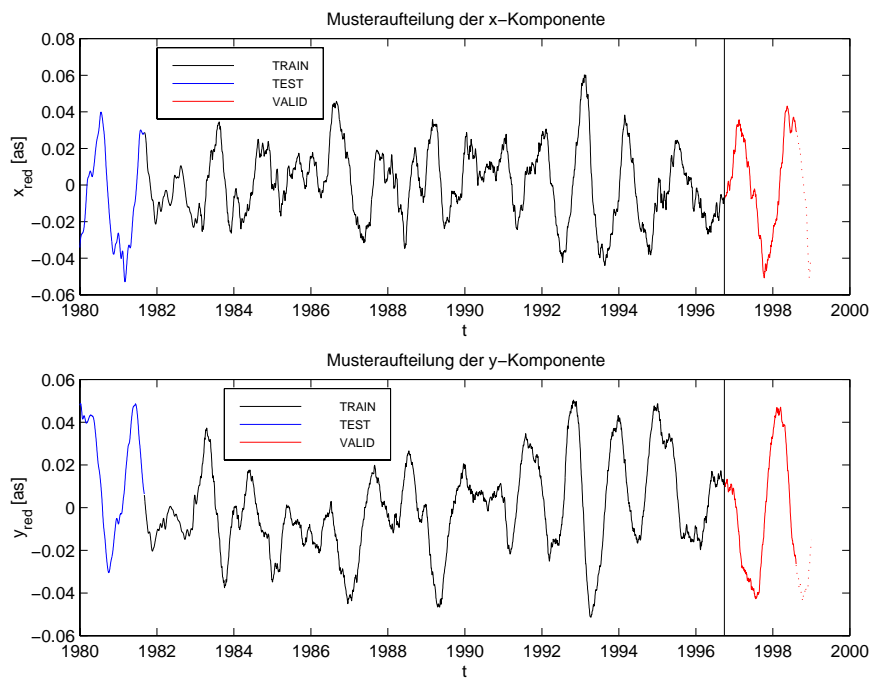


Abbildung 5.2: Musteraufteilung der Polbewegung

übrigen 90 Prozent werden wiederum die ersten 10 Prozent als Testmuster abgestellt (siehe Abbildung 5.2¹). Die maximale Anzahl an versteckten Neuronen ergibt sich damit zu 110.

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	3	
Musteraufteilung	TRAIN+TEST: 90% davon TRAIN: 90% VALID: 10%	6116 Muster 5504 Muster 680 Muster
Eingabemuster	y [1 2 3 5 10]	n=5
Ausgabemuster	y [1 2 3 5 10]	m=5
Netzarchitektur	feedforward: variabel	
Verknüpfung	vollständig	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.8$
Updatefunktion	Topological_Order	
Initialisierungsfunktion	Randomize_Weights	Intervall [-1; 1]
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	zufällig	

Der Trainingsvorgang wird mit einer Lernrate von $\eta = 0.8$ gestartet. Im Laufe des Trainings wird diese aus oben genannten Gründen schrittweise verkleinert. Der Netzausgabefehler des 10. Vorhersagetages nach 10 000 Trainingsiterationen ist in folgender Tabelle zusammengestellt:

Netztopologie	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
5-110-5	4.06	3.62	4.57
5-55-5	4.08	3.64	4.59
5-10-5	4.05	3.64	4.59
5-55-55-5	4.00	3.66	4.61
5-25-25-5	3.95	3.60	4.54
5-10-10-5	3.94	3.59	4.53
5-35-35-35-5	3.91	3.77	4.69
5-15-15-15-5	3.96	3.68	4.84

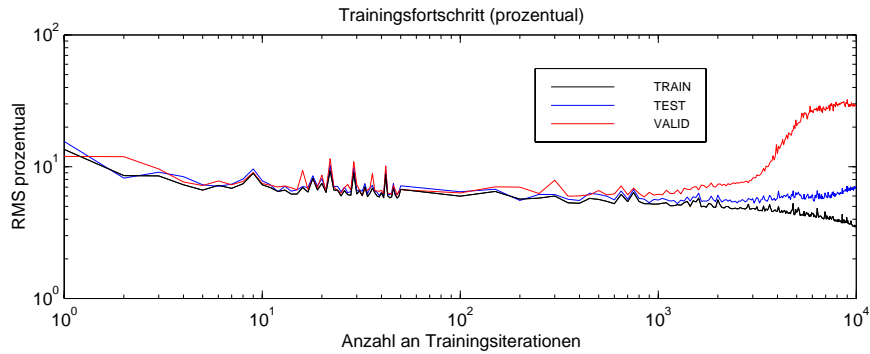
Die RMS-Fehler aller Musterteilmengen verändern sich bei unterschiedlichen Netztopologien nur geringfügig. Die Validierungsmuster besitzen noch einen relativ hohen Wert. Es ist jedoch möglich, dass die neuronalen Netze nach 10 000 Iterationen bereits übertrainiert sind. Deshalb wird der RMS-Fehler der Testmuster während des Lernens überwacht und nach jeder 10. Iteration mit dem bisherigen minimalen Wert verglichen. Der Netzzustand mit den aktuellen Gewichtswerten wird dann an der Stelle des Minimums gespeichert. Bei den vorliegenden Netzen stellte sich allerdings keine wesentliche Verbesserung ein:

Netztopologie	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
5-110-5	9500	4.08	3.60	4.56
5-55-5	9840	4.10	3.62	5.01
5-10-5	9510	4.08	3.63	4.58
5-55-55-5	7100	4.03	3.63	4.59
5-25-25-5	8030	3.99	3.59	4.54
5-10-10-5	7940	3.96	3.57	4.50
5-35-35-35-5	3400	4.00	3.70	4.60
5-15-15-15-5	6100	4.01	3.66	4.80

Wie zu erwarten ist, werden die Fehler der Trainingsmuster durchschnittlich größer, während die meisten Validierungsmuster kleinere Fehler besitzen als nach 10 000 Iterationen. Trotz der hier nur gering ausfallenden Verbesserung ist es nicht ausreichend, das Training nach einer zu Beginn festgelegten Anzahl an Iterationen abzubrechen.

¹Dargestellt ist jeweils der erste Eingabewert der Muster.

Wie wichtig es ist, den RMS-Fehler der Testmuster zu überwachen und als weiteres Abbruchkriterium heranzuziehen, zeigt folgende Graphik, die einen typischen Trainingsverlauf zeigt:



Während sich der Fehler der Trainingsmuster mit zunehmenden Trainingsiterationen verkleinert, wachsen die Fehler der Test- und Validierungsmuster nach einem bestimmten Zeitpunkt an. Leider sind die Zeitpunkte der Minima von Test- und Validierungsmuster nicht exakt gleich. Dadurch erreichen wir zwar fast nie die bestmögliche Vorhersage, trotzdem ist dies eine sehr gute Methode, ein brauchbares Ergebnis zu erhalten.

Aufgrund der relativ hohen Fehler des Validierungsdatensatzes können wir annehmen, dass im verwendeten Eingabemuster nicht die nötige Information für eine Vorhersage von bis zu 10 Tagen enthalten ist. Außerdem ist deutlich zu erkennen, dass in dieser Konfiguration mehrschichtige Netze offensichtlich schneller lernen als Netze mit nur einer versteckten Schicht, da das Minimum nach relativ wenigen Iterationen erreicht wird.

Konfiguration 2

Um eine besser geeignete Mustereinteilung für die Vorhersage der kurzfristigen Schwankungen zu finden, vergrößern wir den Einzugsbereich der Eingabemuster und konzentrieren uns bei der Netzausgabe nur auf den 10. Vorhersagetag:

Einstellmöglichkeit	Einstellung	Ergänzung
Eingabemuster	y [1 10 20 30 40 50 60 70 80 90 100]	$n=11$
Ausgabemuster	y [10]	$m=1$

Da bei den Erdrotationsparametern nicht modellierbare Schwankungen mit Perioden bis zu 100 Tagen auftreten, legen wir den Einzugsbereich auf 100 Tage fest, um diese berücksichtigen zu können. Da der zehnte Tag als Vergleichskriterium herangezogen werden soll, ist der zeitliche Abstand zwischen erstem und zweitem Eingabeneuron mit neun Tagen kleiner als der Abstand der darauffolgenden Neuronen. Dies sollte allerdings nicht weiter stören. Auch für diese Konfiguration werden verschiedene Topologien miteinander verglichen:

Netztopologie	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
11-90-1	7830	3.84	3.94	5.48
11-45-45-1	3750	3.86	4.21	5.45
11-30-1	9840	3.80	4.00	5.15
11-15-15-1	500	3.99	4.03	5.33

Anhand des sehr hohen RMS-Fehlers der Validierungsmuster können wir folgern, dass auch dieses Eingabemuster nicht optimal gewählt ist. Vergleichen wir mit Konfiguration 1, so scheint die Information der letzten 10 Tage ausschlaggebend zu sein und darf deshalb nicht im Eingabemuster fehlen. Sehr wahrscheinlich ist auch, dass die vergangenen Werte der Erdrotation mit zunehmendem Alter an Bedeutung für die Vorhersage verlieren. Um dies zu berücksichtigen, wählen wir eine neue Mustereinteilung:

Konfiguration 3

Neben Berücksichtigung der letzten 10 Tage erhalten die weiter zurückliegenden Tage immer weniger Einfluss, indem der relative Abstand zweier Neuronen in der Schrittweite von 2er-Potenzen zunimmt:

Einstellmöglichkeit	Einstellung	Ergänzung
Eingabemuster	y [1...10 12 16 24 40 72 136]	n=16
Ausgabemuster	y [10]	m=1

Hierbei sind teilweise deutliche Verbesserungen zu beobachten:

Netztopologie	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
16-57-1	8410	3.71	3.70	4.38
16-28-1	8330	3.63	3.71	4.42
16-28-28-1	7800	3.52	3.84	5.74
16-20-10-1	5440	3.68	3.75	5.75
16-8-8-1	4430	3.61	3.65	4.85
16-8-4-1	4170	3.57	3.55	4.41
16-4-2-1	3980	3.66	3.73	5.05

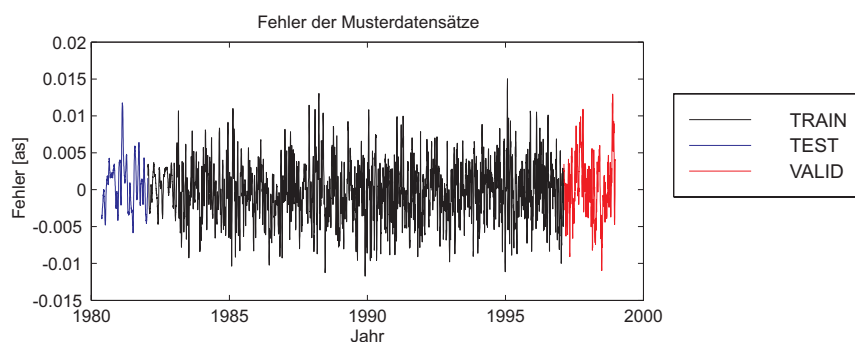
Am besten ist wohl das Ergebnis der vorletzten Netztopologie in der Tabelle mit zwei versteckten Schichten und insgesamt 12 versteckten Neuronen zu bewerten. Zwar ist der Validierungsfehler hier geringfügig größer als bei der Topologie an oberster Stelle, aber es wurden nur etwa die Hälfte an Trainingsiterationen benötigt. Eine größere Anzahl an versteckten Neuronen wirkt sich allerdings negativ auf das Ergebnis aus.

Konfiguration 4

Der große Genauigkeitsunterschied zwischen Test- und Validierungsmustern lässt vermuten, dass zwar die Testmuster vom Trainingsdatensatz gut, die Validierungsmuster allerdings sehr schlecht repräsentiert werden. Das Hinzufügen weiterer Trainingsdaten durch Einbeziehen der Polbewegung in den Jahren 1980 bis 1983 soll zeigen, ob in diesem Zeitraum Information enthalten ist, die die Vorhersage positiv beeinflusst. Tatsächlich ist das Ergebnis wesentlich besser:

Netztopologie	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
16-8-4-1	3580	3.84	3.39	3.95

Interessant ist an dieser Stelle der Fehlerplot der einzelnen Musterdatensätze:



Deutlich ist zu erkennen, dass der Fehler bis etwa zum Jahr 1983 geringer ausfällt als im übrigen Zeitraum. An dieser Stelle könnten wir vermuten, dass beim Trainieren des Netzes trotz zufälliger Musterpräsentation diejenigen Muster besser repräsentiert werden, die am Anfang des Datensatzes (TRAIN-Datei) stehen. Nach Invertieren der Reihenfolge der Muster innerhalb der Musterdatensätze ergab sich allerdings ein ähnliches Bild. Die Ursache ist wohl eher in der unterschiedlichen Glättung der Daten zu suchen. Vor dem Jahre 1983 waren weniger hochfrequente Anteile enthalten. Dies ist mit einem gleitenden Mittel vergleichbar, von dem die Netzausgabe natürlich weniger abweicht als von sehr hochfrequenten Schwankungen.

Konfiguration 5

Bisher waren die Schichten aller verwendeten Netze vollständig verknüpft. Die Auswirkung zusätzlicher shortcut-Verbindungen auf das in Konfiguration 4 verwendete Netz zeigt folgende Tabelle:

Netztopologie	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
16-8-4-1, shortcut	3920	3.76	3.14	3.78

Dies ist gegenüber Konfiguration 4 eine signifikante Genauigkeitssteigerung. Durch die shortcut-Verbindungen wird dem Netz ermöglicht, die Eingabewerte direkt als zusätzliche Eingangsinformation an den Ausgabeneuronen anzulegen.

Konfiguration 6

Um die kurzfristigen Schwankungen der nächsten 10 Tage vorherzusagen, könnten wir analog zu obigem Vorgehen für jeden Vorhersagetag ein individuelles optimales Netz suchen. Dies wäre jedoch ein sehr unwirtschaftliches Vorgehen. Außerdem ist an einen späteren praktischen Einsatz zu denken, bei dem nach Möglichkeit nur noch ein neuronales Netz trainiert werden soll. Deshalb ist es naheliegend, alle Vorhersagetage durch insgesamt zehn Ausgabeneuronen zu repräsentieren.

Dazu werden drei verschiedene Netze untersucht. Das Ergebnis fällt allerdings sehr unterschiedlich aus:

Netztopologie	Verknüpfung	Min. RMS_{TEST} bei Iteration	RMS_{TRAIN}^{10} [mas]	RMS_{TEST}^{10} [mas]	RMS_{VALID}^{10} [mas]
16-20-20-10	vollständig	1590	3.77	3.41	4.24
16-20-20-10	shortcut	3480	3.49	2.88	4.09
16-8-4-10	shortcut	4060	3.69	3.24	3.89
16-4-2-10	shortcut	4390	3.76	3.28	4.01

Neben der Tatsache, dass sich die vollständige Vernetzung auch hier als wenig vorteilhaft herausstellt, ist erstaunlich, dass dieselbe Anzahl an versteckten Neuronen die besten Ergebnisse liefert, wie bei der separaten Vorhersage des 10. Tages (vgl. Konfiguration 5). Die Fehler dieses Netzes mit 12 versteckten Neuronen zeigt die erste Zeile folgender Übersicht:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
Gemeinsame Vorhersage RMS_{VALID} [mas]	0.52	0.88	1.24	1.69	2.09	2.46	2.82	3.17	3.52	3.89
(16-8-4-10, shortcut)										
Einzelvorhersage RMS_{VALID} [mas]	0.40	0.77	1.22	1.67	2.10	2.49	2.90	3.08	3.48	3.78
(16-8-4-1, shortcut)										

Der geringfügige Genauigkeitsverlust beim 10. Vorhersagetag gegenüber Konfiguration 5 ist durchaus akzeptabel, wenn wir uns vor Augen halten, dass mit nur einem Netz die nächsten 10 Tage gleichzeitig vorhergesagt werden können. Um die Vorhersagetage 1 bis 9 trotzdem mit Konfiguration 5 vergleichen zu können, wird für jeden dieser Tage eine eigene Musterkonfiguration erstellt, die neben den bekannten 16 Eingabeneuronen nur jeweils ein Ausgabeneuron besitzen. Ansonsten entsprechen die Einstellungen der von Konfiguration 5. Damit ist eine separate Vorhersage aller 10 Tage analog zu Konfiguration 5 möglich. Die Ergebnisse sind in obiger Tabelle in der zweiten Zeile zu finden. Für die meisten Tage besitzt die Einzelvorhersage geringe Vorteile gegenüber einer gemeinsamen Prädiktion, allerdings sind die Unterschiede nur sehr gering. Deshalb ist es fraglich, ob signifikante Unterschiede zwischen beiden Methoden existieren.

Konfiguration 7

Mit den gefundenen optimalen Einstellungen aus den Konfigurationen 5 und 6 wird nun die y-Komponente der Polbewegung vorhergesagt. Der RMS-Fehler der Validierungsmuster an der Stelle des minimalen RMS-Fehlers der Testmuster dient wiederum der Analyse:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
Gemeinsame Vorhersage	0.36	0.64	0.89	1.16	1.43	1.67	1.87	2.06	2.25	2.44
<i>RMS_{VALID}</i> [mas]	(16-8-4-10, shortcut)									
Einzelvorhersage	0.30	0.60	0.86	1.15	1.45	1.70	1.92	2.10	2.38	2.59
<i>RMS_{VALID}</i> [mas]	(16-8-4-1, shortcut)									

Hier ist die direkte Einzelvorhersage der meisten Tage sogar etwas schlechter als die gemeinsame Vorhersage. Der nur geringe Genauigkeitsvorteil der Einzelvorhersage der x-Komponente gegenüber einer gemeinsamen Vorhersage ist anscheinend nicht signifikant. Deshalb ist die gemeinsame Vorhersage aller zehn Tage einer Einzelvorhersage in jedem Fall vorzuziehen. Es fällt auch auf, dass der Vorhersagefehler der y-Komponente allgemein kleiner ist als bei der x-Komponente. Das liegt wahrscheinlich daran, dass die gemessenen Polkoordinaten in y in der Regel genauer sind als in x.

Konfiguration 8

Eine weitere Möglichkeit, die Polbewegung vorherzusagen, besteht darin, beide Komponenten an die Eingabeneuronen eines gemeinsamen Netzes anzulegen. Dazu erhält das 1. Eingabeneuron den 1. Wert der x-Komponente, das 2. den ersten Wert der y-Komponente, das 3. den 2. Wert der x-Komponente und so weiter. Die Ausgabeneuronen werden dementsprechend belegt. Zwei Netze werden auf ihre Eignung hin untersucht. Da sich die Anzahl der Ein- und Ausgabeneuronen auf 32 und 20 verdoppelt, erscheint es sinnvoll, auch die Anzahl der versteckten Neuronen zu erhöhen. Das Ergebnis bestätigt diese Vermutung:

Vorhersagetag		1	2	3	4	5	6	7	8	9	10
32-8-4-20, shortcut	x	0.49	0.98	1.52	2.17	3.00	2.55	4.29	5.01	5.66	6.22
<i>RMS_{VALID}</i> [mas]	y	0.34	0.60	0.91	1.20	1.81	2.18	2.59	2.98	3.41	3.52
32-16-8-20, shortcut	x	0.47	0.92	1.47	2.16	2.97	3.60	4.35	5.08	5.74	6.31
<i>RMS_{VALID}</i> [mas]	y	0.35	0.63	0.92	1.31	1.75	2.16	2.58	2.95	3.30	3.60
32-32-16-20, shortcut	x	0.34	0.67	1.08	1.58	2.18	2.63	3.18	3.71	4.19	4.61
<i>RMS_{VALID}</i> [mas]	y	0.23	0.46	0.68	0.95	1.28	1.58	1.89	2.16	2.41	2.63

Die Netze mit gleicher und doppelter Anzahl versteckter Neuronen wie in Konfiguration 6 erzielen ein wesentlich schlechteres Ergebnis als das Netz mit der vierfachen Anzahl, d.h. mit 48 versteckten Neuronen. Die Vorhersage der nächsten 4 bzw. 6 Tage gelingt für die x- bzw. y- Komponente sogar besser als bei der getrennten Vorhersage beider Komponenten (fettgedruckte Werte in der Tabelle). Die x-Komponente enthält also komplementäre Information für die kurzfristige Vorhersage der y-Komponente und umgekehrt. Das Maß dieser Information geht allerdings mit größer werdendem Vorhersagezeitraum zurück.

Die endgültige Prädiktion wird daher aufgespalten in eine Vorhersage der kommenden fünf Tage, bei der die gemeinsame Vorhersage beider Komponenten genutzt wird und in eine Vorhersage der Tage 6 bis 10, bei der die ersten 10 Tage der x- und die y-Komponenten getrennt behandelt werden.

Zwischenergebnis

Zur Vorhersage der kurzfristigen Schwankungen der Polbewegung werden folgende Konfigurationen verwendet:

Vorhersagetag	Neuronales Netz	Topologie	Verknüpfung	Art
1 bis 5	feedforward	32-32-16-20	short cut	x, y gemeinsam 1 bis 10 gemeinsam
5 bis 10	feedforward	16-8-4-10	short cut	x, y getrennt 1 bis 10 gemeinsam

Um auf einen gemeinsamen Genauigkeitswert zu kommen, errechnen wir uns den RMS-Fehler der Polbewegung aus den Fehlern der Einzelkomponenten:

$$RMS^x = \sqrt{\frac{\sum_{j=1}^{n^x} (o_j^x - t_j^x)^2}{n^x}} \quad \text{bzw.} \quad RMS^y = \sqrt{\frac{\sum_{j=1}^{n^y} (o_j^y - t_j^y)^2}{n^y}} \quad (5.1)$$

mit $n^x = n^y =: n$ folgt :

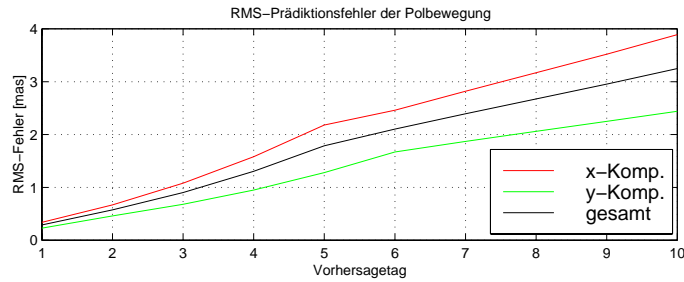


Abbildung 5.3: RMS-Prädiktionsfehler der einzelnen Komponenten und der Gesamtfehler

$$\begin{aligned}
 RMS^{pol} &= \sqrt{\frac{\sum_{j=1}^n (o_j^x - t_j^x)^2 + \sum_{j=1}^n (o_j^y - t_j^y)^2}{2n}} \\
 &= \sqrt{\frac{(RMS^x)^2 + (RMS^y)^2}{2}} \tag{5.2}
 \end{aligned}$$

Die erreichten Ergebnisse sind in Abbildung 5.3 graphisch dargestellt und in folgender Tabelle noch einmal übersichtlich zusammengefasst:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
RMS^x [mas]	0.34	0.67	1.08	1.58	2.18	2.46	2.82	3.17	3.52	3.89
RMS^y [mas]	0.23	0.46	0.68	0.95	1.28	1.67	1.87	2.06	2.25	2.44
RMS^{pol} [mas]	0.29	0.57	0.90	1.30	1.79	2.10	2.39	2.67	2.95	3.25

In Abbildung 5.4 ist die Gegenüberstellung von tatsächlichem Verlauf der reduzierten Polbewegung und Vorhersage des 10. Tages zu sehen. Auffällig ist vor allem bei der x-Komponente, dass die Vorhersage dem tatsächlichen Verlauf teilweise etwas "hinterherläuft". Nach näherer Untersuchung können wir feststellen, dass die Größe des Versatzes 10 Tage nicht überschreitet. Ist das Netz nicht in der Lage, eine Vorhersage des 10. Tages zu treffen, so verkürzt es anscheinend selbstständig den Vorhersagezeitraum. Dies geht sogar soweit, dass es bei sehr schwierigen Verläufen für die Vorhersage schlicht den Funktionswert des letzten Tages annimmt. In diesem Fall beträgt der Versatz 10 Tage. Dieses Vorgehen mag im ersten Augenblick falsch erscheinen, ist aber eine Möglichkeit, auch bei unzureichender Information eine Vorhersage zu treffen.

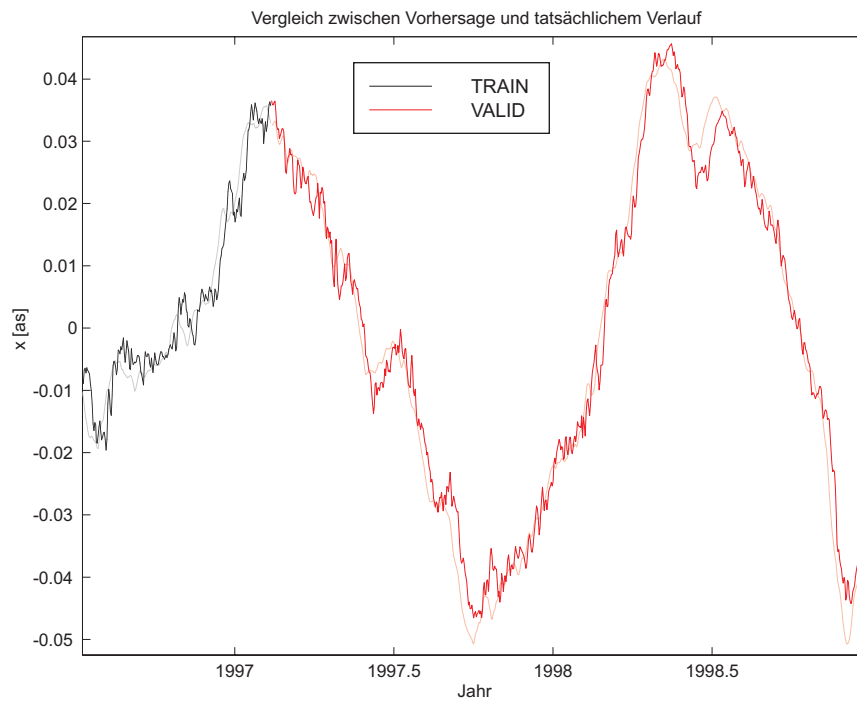
Vorhersage der Tage 11 bis 40

Die gewonnenen Kenntnisse werden nun auf eine weiterführende Vorhersage angewandt. Da die getrennte Vorhersage der x- und y-Komponente für die Tage 5 bis 10 einer gemeinsamen überlegen ist, nutzen wir auch hier ein feedforward-Netz mit der Topologie 16-8-4-10. Die Vorhersage der Tage 11 bis 40 wird also in 3 Teilaufgaben zerlegt, bei denen jeweils 10 Tage prädiziert werden.

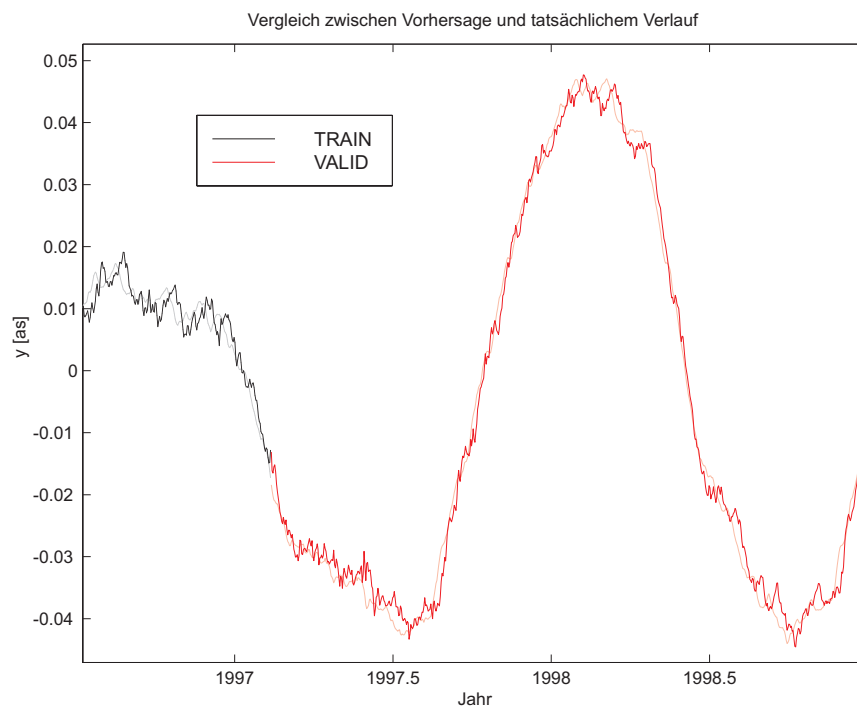
Werden zur Bestimmung der Verbindungsgewichte des Netzes, das die Tage 1 bis 10 der x-Komponente vorhersagt, noch 4060 Trainingsiterationen benötigt, um das Fehlerminimum der Testmuster zu erreichen, so sinkt diese Zahl mit steigendem Vorhersagezeitraum: Für eine Vorhersage der Tage 11 bis 20 sind nur noch 360, für den nächsten Abschnitt der Tage 21 bis 30 lediglich 110, und für die Tage 31 bis 40 nicht mehr als 90 Trainingszyklen notwendig. Analoges gilt für die Vorhersage der y-Komponente. Erklären können wir uns diesen Effekt, wenn wir uns das Trainieren einer Funktion als inversen Tiefpassfilter vorstellen. Innerhalb der ersten Trainingsiterationen versucht das Netz, die grobe Struktur, d.h. die tiefen Frequenzen der zu approximierenden Funktion zu repräsentieren. Mit fortlaufendem Training können immer mehr Feinheiten (hohe Frequenzen) vom Netz angelernt werden. Je weiter der Vorhersagetag in der Zukunft liegt, desto mehr gewinnen tiefere Frequenzen an Bedeutung für die Vorhersage und desto eher ist die Abbruchbedingung des minimalen Fehlers des Testdatensatzes erfüllt.

In Abbildung 5.5 sind die endgültigen Ergebnisse der Vorhersagetage 1 bis 40 graphisch veranschaulicht. Folgende Tabelle zeigt dazu einige numerische Werte:

Vorhersagetag	5	10	15	20	25	30	35	40
RMS^{pol} [mas]	1.79	3.25	4.70	6.28	7.78	8.89	10.14	10.96



(a) x-Komponente



(b) y-Komponente

Abbildung 5.4: Vergleich zwischen Vorhersage des 10. Tages (dunkle Farbe) und tatsächlichem Verlauf (helle Farbe)

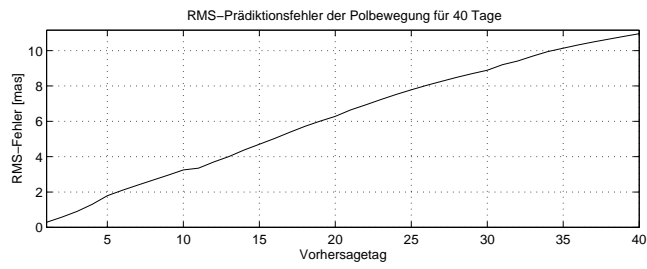


Abbildung 5.5: Prädiktionsfehler der Polbewegung

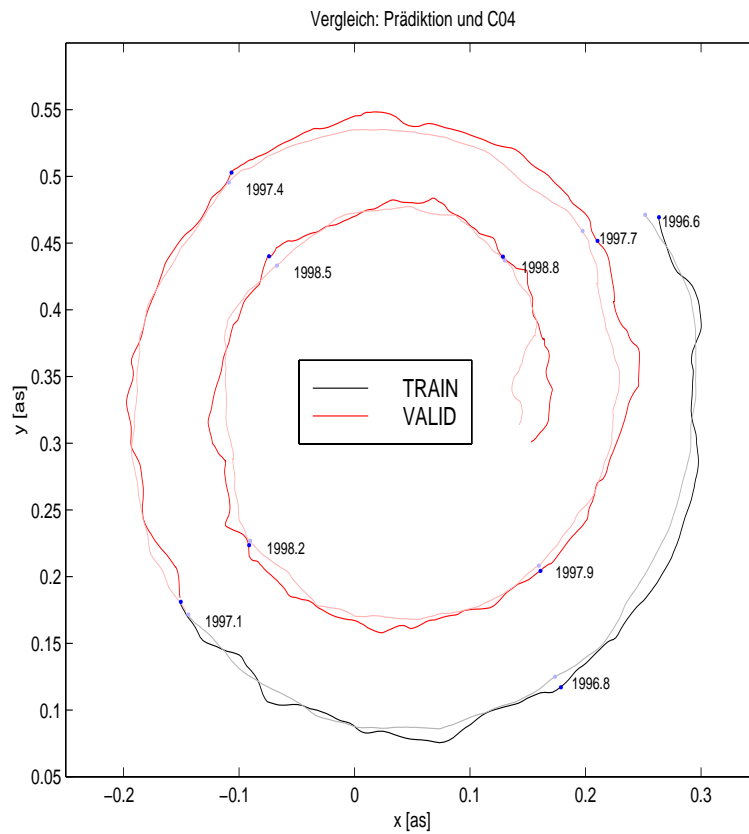


Abbildung 5.6: Vergleich der Prädiktion 40 Tage in die Zukunft (dunkle Farbe) mit der ursprünglichen Datenreihe C04 (helle Farbe)

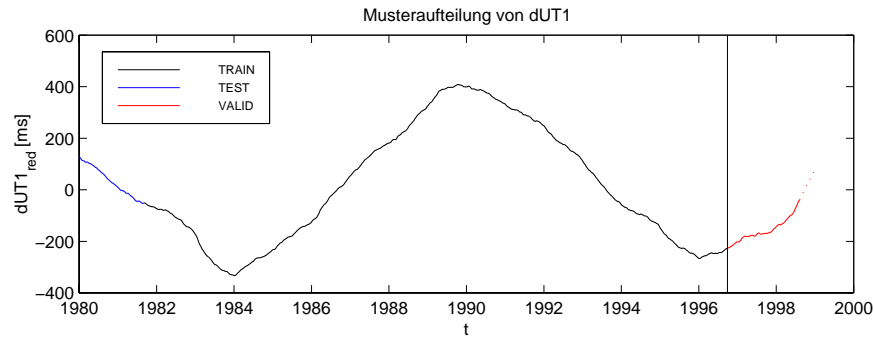


Abbildung 5.7: Musteraufteilung von $\Delta UT1$

Die endgültige Vorhersage des 40. Tages der Polbewegung nach Addition der im Vorfeld reduzierten Einflüsse ist in Abbildung 5.6 zu sehen.

5.1.2 Vorhersage von $\Delta UT1$

Das in Kapitel 5.1.1 beschriebene schrittweise Vorgehen zur Prädiktion der Polbewegung wird nun analog auf $\Delta UT1$ angewendet. Deshalb beschränkt sich die folgende Beschreibung auf die charakteristischen Besonderheiten bezüglich der Vorhersage von $\Delta UT1$.

Die Musteraufteilung in Test-, Trainings- und Validierungsmuster zeigt Abbildung 5.7.

Konfiguration 1

Die erste Eigenheit fällt bereits während des ersten Trainingslaufes auf: Im Gegensatz zum Lernen der Polbewegung benötigen die Netze zum Trainieren von $\Delta UT1$ wesentlich mehr Iterationen, um auf ein brauchbares Fehlermaß zu kommen.

Verwenden wir das neuronale Netz, das bei der Vorhersage der Polbewegung die besten Erfolge erzielte, wird das Minimum erst nach 289 090 Iterationen erreicht. Dies entspricht einer Rechenzeit² von ca. 27 Stunden. Der hohe Anteil an unreduzierten Restschwingungen in $\Delta UT1$ könnte Grund für den erhöhten Trainingsbedarf sein. Vor allem die langperiodische Schwingung muss zunächst vom Netz repräsentiert werden, bevor die Feinheiten, die für eine Vorhersage der kurzfristigen Schwankungen ausschlaggebend sind, im Netz gespeichert werden können. Eine weniger starke a priori Reduktion der Daten muss also vom Netz durch längeres Trainieren kompensiert werden.

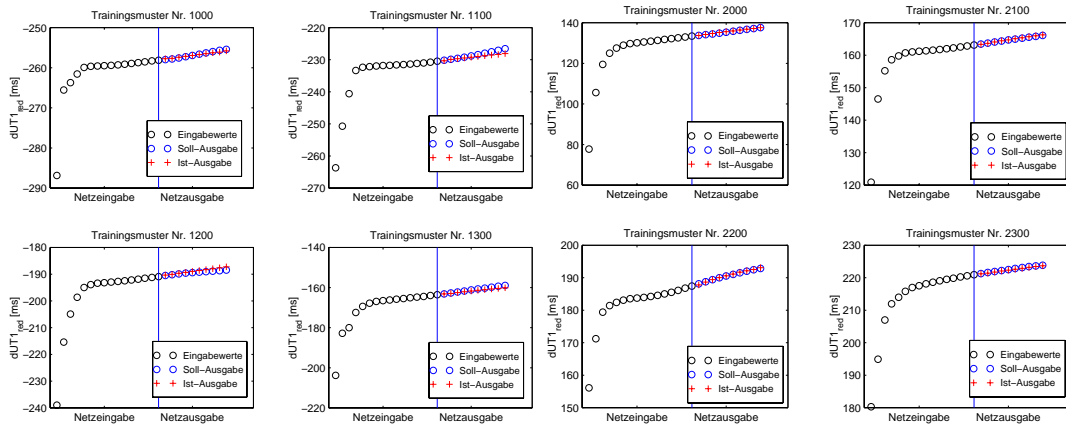
Das erzielte Ergebnis ist dann allerdings bereits sehr gut. In Abbildung 5.8 sind exemplarisch einige Muster mit ihren Netzein- und Ausgabewerten dargestellt. In den meisten Fällen steht die Vorhersagequalität der Validierungsmuster derjenigen der Trainingsmuster in nichts nach:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
RMS_{TRAIN} [ms]	0.23	0.24	0.30	0.39	0.50	0.62	0.75	0.88	1.02	1.17
RMS_{TEST} [ms]	0.13	0.20	0.25	0.34	0.46	0.57	0.71	0.86	1.01	1.18
RMS_{VALID} [ms]	0.33	0.25	0.29	0.39	0.49	0.62	0.75	0.90	1,05	1.22

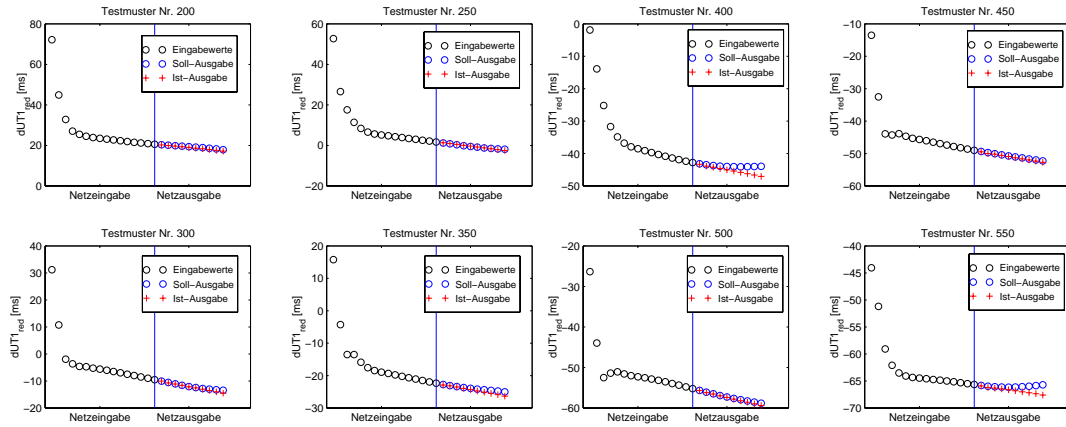
Wir erhalten demzufolge nur dann eine sehr gute Genauigkeit, wenn das Netz sehr lange trainiert wird. Das Ziel ist es nun, eine ähnlich hohe Genauigkeit mit weniger Trainingszyklen zu erreichen.

Im Gegensatz zur Vorhersage der Polbewegung, bei der bereits nach wenigen Iterationen das angestrebte Minimum erreicht wird, ist es in diesem Fall sinnvoll, neben verschiedenen Netztopologien auch unterschiedliche Lernfunktionen einzusetzen, um das Training zu optimieren. Da es zeitlich nicht vertretbar ist, alle Netze bis zum Minimum zu trainieren, legen wir als Abbruchbedingung eine maximale Iterationszahl von 30 000 fest. Die zu diesem Zeitpunkt auftretenden RMS-Fehler werden dann als Vergleichskriterium herangezogen, um die ökonomischste Konfiguration zu finden.

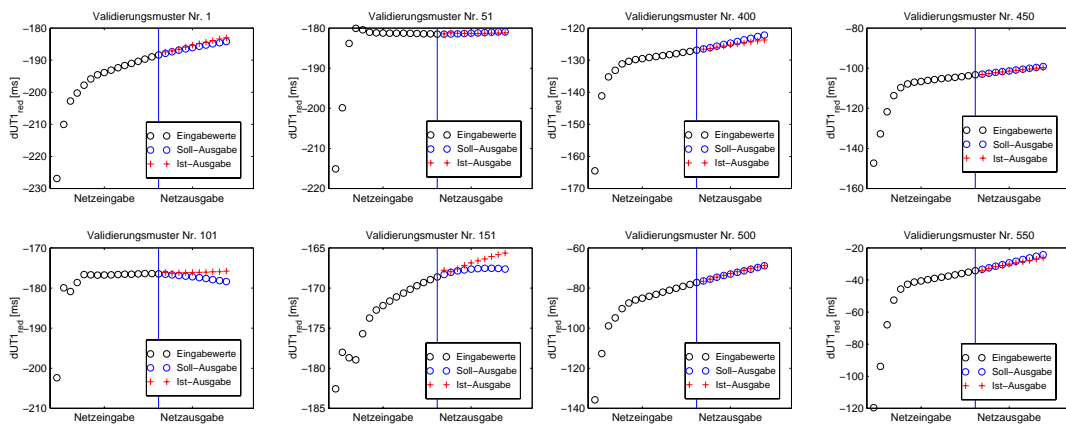
²Trainieren des Netzes auf Pentium III mit 500 MHz Taktfrequenz.



(a) Trainingsmuster



(b) Testmuster



(c) Validierungsmuster

Abbildung 5.8: Darstellung einiger $\Delta UT1$ -Musterpaare mit Netzein- und ausgabe

Konfiguration 2

Vermindern oder erhöhen wir die Anzahl der versteckten Neuronen zu stark, so bedingt dies einen Genauigkeitsverlust. Zu wenige Neuronen können die Funktion nicht mehr repräsentieren, wohingegen zu viele Neuronen zu lange Trainingszeiten benötigen.

Die Ergebnisse nach Verwendung unterschiedlicher Netztopologien und leichter Variation der Anzahl an versteckten Neuronen unterscheiden sich nur geringfügig voneinander. Diese Unterschiede lassen keine qualitative Bewertung der Topologien zu, da sie nicht signifikant sind. Sie liegen vielmehr in dem Schwankungsbereich, in dem sich die Fehlerkurven zwischen zwei Trainingsiterationen bewegen. Daher verwenden wir auch im Weiteren die bewährte Netztopologie aus Kapitel 5.1.1 mit 8 und 4 versteckten Neuronen.

Konfiguration 3

Der Einsatz neuer Lernfunktionen (siehe Kapitel 3.1.1) hingegen wirkt sich im Vergleich zur Standard-Backpropagation (*Std_Backpropagation*) sehr deutlich auf das Ergebnis aus. Die Lernverfahren *BackpropWeightDecay* und *BackpropBatch* führen erstaunlicherweise zu einem wesentlich schlechteren bis unbrauchbaren Ergebnis.

Die *Quickprop*-Lernregel versagt gänzlich: Der Netzausgabefehler ist selbst nach sehr vielen Trainingsiterationen und unterschiedlicher Parameterwahl größer als der Fehler nach der Netzinitialisierung.

Die Erweiterungen der Backpropagation-Lernregel um einen Momentum-Term und die Flat-Spot Elimination bewirken dagegen eine erhebliche Verkürzung der Lernzeit des Netzes. Die Besonderheit dieses sog. *Enhanced Backpropagation* (*BackpropMomentum*) soll im Folgenden kurz vorgestellt werden:

Die Formeln für Standard-Backpropagation (siehe Kapitel 2.2.2, Formeln 2.18 und 2.19) werden um die zwei Parameter μ und c erweitert:

$$\Delta w_{ij}(t+1) = \eta o_i \delta_j + \mu \Delta w_{ij}(t) \quad (5.3)$$

$$\delta_j = \begin{cases} (f'_{act}(net_j) + c)(t_j - o_j) & \text{falls } j \text{ ein Ausgabeneuron ist} \\ (f'_{act}(net_j) + c) \sum_k (\delta_k w_{jk}) & \text{falls } j \text{ ein verstecktes Neuron ist} \end{cases} \quad (5.4)$$

Dabei ist

- μ der Parameter des Momentum-Terms. Er legt den konstanten, relativen Einfluss des "alten" Gewichtes $w_{ij}(t)$ auf die Berechnung des "neuen" Gewichtes $\Delta w_{ij}(t+1)$ fest. Diese Korrektur wird zur herkömmlichen Gewichtsänderung der Standard-Backpropagation addiert. Dadurch kann ein Oszillieren des Netzausgabefehlers während des Trainierens vermieden werden. Positiven Einfluss hat dies bei sehr steilen Schluchten in der Fehlerfläche (siehe Abbildung 5.9 (a)). Da dort der Gradient einen sehr großen Wert besitzt, ist auch die Gewichtsänderung dementsprechend groß. Handelt es sich um eine sehr enge Schlucht, so kann es durchaus vorkommen, dass durch die Gewichtsänderung ein Sprung auf die gegenüberliegende Seite erfolgt. Ist dort der Gradient, der jetzt natürlich in die andere Richtung zeigt, ähnlich groß, so springen wir wieder zurück auf die ursprüngliche Position. Das globale Minimum kann somit nicht erreicht werden.
- c der Wert der Flat-Spot Elimination. Dieser konstante Parameter wird zur Ableitung der Aktivierungsfunktion addiert. Dadurch wird dem Netz ermöglicht, auch flache Plateaus auf der Fehleroberfläche, in denen die Ableitung $f'_{act}(net_j)$ sehr kleine Werte besitzt, zu überwinden (siehe Abbildung 5.9 (b)). Ohne Addition dieses Wertes würde Backpropagation stagnieren bzw. sehr viele Iterationsschritte benötigen, da die Gewichtsänderung proportional zum Betrag des Gradienten - der dann ungefähr einen Wert von 0 hätte - vollzogen wird. In einem solchen Fall könnten wir außerdem nicht unterscheiden, ob wir uns in einem Minimum oder auf einem flachen Plateau der Fehlerfunktion befinden. Diese Erweiterungen führen zu einem Beschleunigen (Erhöhen der Gewichtsänderung) in weiten Plateaus und zu einem Abbremsen in sehr unruhigen Fehlerflächen.

Für die Vorhersage von $\Delta UT1$ setzen wir die Parameter μ und c auf die von Zell (1994) empfohlenen Werte von 0.6 und 0.1. Das Ergebnis ist beeindruckend: Bereits nach nur 30 000 Trainingsiterationen liegen die Netzausgabefehler der Musterdatensätze deutlich unter den Werten, die Standard-Backpropagation nach insgesamt 290 000 Iterationen liefert:

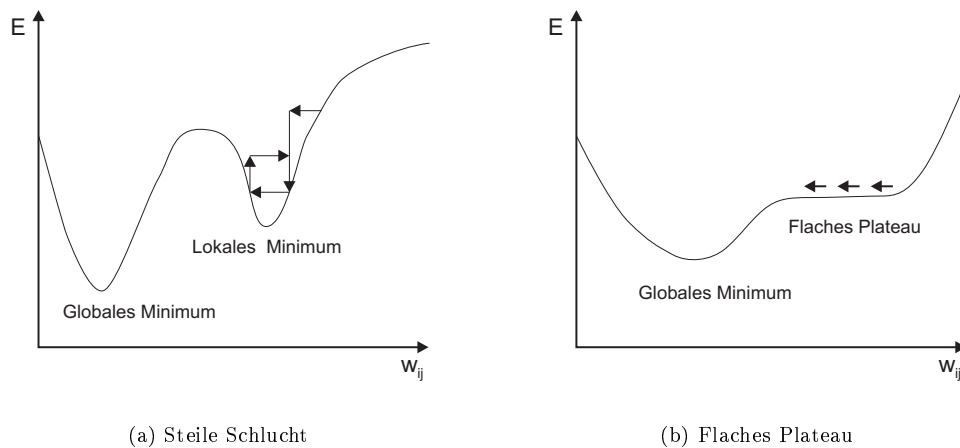


Abbildung 5.9: Vorteile des Momentum-Terms und der Flat-Spot Elimination

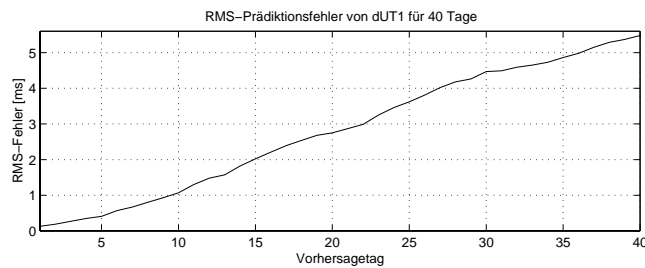


Abbildung 5.10: Prädiktionsfehler von $\Delta UT1$

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
RMS_{TRAIN} [ms]	0.19	0.22	0.30	0.38	0.49	0.59	0.69	0.82	0.94	1.07
RMS_{TEST} [ms]	0.12	0.18	0.26	0.33	0.45	0.56	0.65	0.77	0.90	1.03
RMS_{VALID} [ms]	0.13	0.19	0.27	0.35	0.41	0.57	0.67	0.80	0.93	1.07

Vorhersage der Tage 11 bis 40

Für die Vorhersage der Tage 11 bis 40 wird der Vorhersagezeitraum ebenfalls in drei Teile à 10 Tagen zerlegt. Desweiteren wird für alle Teile die obige Konfiguration verwendet.

Auch hier tritt wieder der Effekt auf, dass Vorhersagen, die weiter in der Zukunft liegen, weniger Trainingsiterationen benötigen: Für die Prädiktion der Tage 11 bis 20 sind noch 27 320 Iterationen nötig, für die Tage 21 bis 30 genügen 10 600. Das Fehlerminimum für die Prädiktion der Tage 31 bis 40 ist bereits nach 230 Trainingsiterationen erreicht. Neben Abbildung 5.10 zeigt nachstehende Tabelle die Ergebnisse:

Vorhersagetag	5	10	15	20	25	30	35	40
$RMS^{\Delta UT1}$ [ms]	0.41	1.07	2.02	2.75	3.62	4.47	4.86	5.48

Zur Veranschaulichung der Fehlerwerte dient die Abbildung 5.11, die einen Vergleich zwischen dem 40. Vorhersagetag und dem tatsächlichen Verlauf der reduzierten Datenreihe zeigt. Wieder ist ein gewisses "Nachlaufen" der Vorhersage zu erkennen.

5.1.3 Vorhersage von ΔLOD

Auch bei der Vorhersage von ΔLOD erweisen sich nach Erprobung von insgesamt 15 verschiedenen Netztopologien und 5 unterschiedlichen Mustereinteilungen wieder die in den beiden vorangegangenen Kapiteln gefundenen Konfigurationen als günstig. Dieser Umstand ist doch sehr erstaunlich, wenn wir bedenken, dass sich die Datenreihen

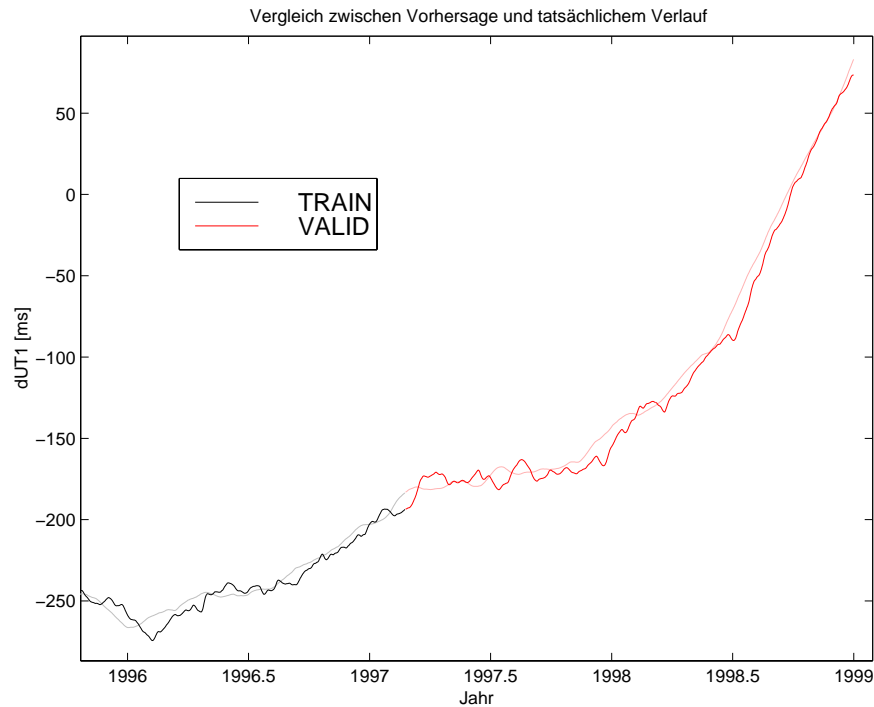


Abbildung 5.11: Vergleich zwischen Vorhersage des 40. Tages (dunkle Farbe) und tatsächlichem Verlauf (helle Farbe)

Polbewegung, $\Delta UT1$ und ΔLOD in ihren Verläufen wesentlich voneinander unterscheiden. Dem Anschein nach besitzen die residualen Datenreihen trotz der offensichtlichen Unterschiede noch sehr starke Gemeinsamkeiten, die sich in erster Linie in den Mustereinteilungen widerspiegeln. Dass sich dann identische optimale Netztopologien herauskristallisieren, ist dagegen weniger verwunderlich, schließlich sind aufgrund der gleichen Mustereinteilungen auch die Anzahl der Eingabe- und Ausgabeneuronen jeweils identisch.

Die verwendeten Einstellungen sind noch einmal übersichtlich zusammengefasst:

Einstellmöglichkeit	Einstellung	Ergänzung
Vorhersagestrategie	3	
Musteraufteilung	TRAIN+TEST: 90% davon TRAIN: 90% VALID: 10%	6116 Muster 5504 Muster 680 Muster
Eingabemuster	$y [1..10 \ 12 \ 16 \ 24 \ 40 \ 72 \ 136]$	$n=16$
Ausgabemuster	$y [1..10]$	$m=10$
Netzarchitektur	feedforward: 16-8-4-10	
Verknüpfung	short cut	
Lernfunktion	Std_Backpropagation	Lernrate $\eta = 0.8$
Updatefunktion	Topological_Order	
Initialisierungsfunktion	Randomize_Weights	Intervall [-1; 1]
Aktivierungsfunktion	logistische Funktion	
Ausgabefunktion	Identität	
Reihenfolge der Musterpräsentation	zufällig	

Die gewählte Musteraufteilung mit 5504 Trainingsmustern, 612 Testmustern und 680 Validierungsmustern ist in Abbildung 5.12 zu sehen.

Nach bereits 2480 Trainingsiterationen ist das Fehlerminimum des Testdatensatzes erreicht. Die Datenreihe ΔLOD konnte im Vorfeld so stark reduziert werden, dass das neuronale Netz vergleichsweise nur sehr wenige Iterationen benötigt, um die Residuen ausreichend genau zu repräsentieren. Vergleichen wir die Spektren der reduzierten Datenreihen von ΔLOD und $\Delta UT1$ (vgl. Kapitel 4.1.4, Abbildung 4.10 und Kapitel 4.2.4, Abbildung 4.15), so

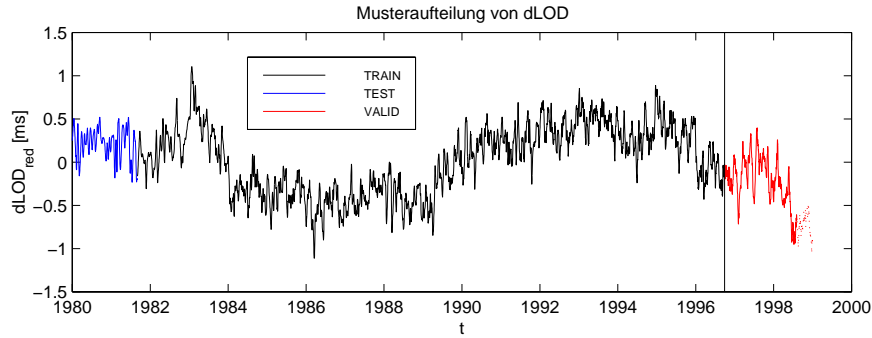


Abbildung 5.12: Musteraufteilung von ΔLOD

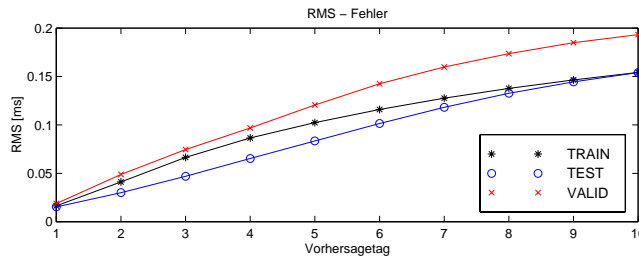


Abbildung 5.13: RMS-Fehler der Vorhersage von ΔLOD für die ersten 10 Tage

leuchten die bei identischen Lernverfahren unterschiedlich langen Trainingsphasen ein.

Das mit dieser Konfiguration erzielte Ergebnis ist in folgender Tabelle zusammengestellt:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
RMS_{TRAIN} [ms]	0.016	0.041	0.066	0.086	0.102	0.116	0.128	0.138	0.146	0.154
RMS_{TEST} [ms]	0.015	0.030	0.047	0.065	0.083	0.101	0.118	0.133	0.144	0.154
RMS_{VALID} [ms]	0.019	0.049	0.074	0.097	0.121	0.142	0.159	0.174	0.184	0.193

Auffällig sind die relativ großen Genauigkeitsunterschiede zwischen Trainings- und Validierungsdaten (vgl. auch Abbildung 5.13). Dies liegt daran, dass in diesem Fall das Fehlerminimum der Testdaten von dem der Validierungsdaten stärker abweicht. Der Trainingsverlauf ist in Abbildung 5.14 zu sehen.

Sehr interessant ist die Gegenüberstellung des vorhergesagten 10. Tages mit dem tatsächlichem Verlauf der reduzierten Datenreihe (vgl. Abbildung 5.15). Das Netz prädiziert um das Jahr 1998 einen sehr auffälligen Peak mit einer Amplitude von ca. 1.2 ms. In den Trainingsdaten ist ein ähnlicher Ausschlag im Jahre 1983 zu sehen (vgl. Abbildung 5.12). Dabei handelt es sich um einen El Nino. Erstaunlicherweise gab es auch 1998 einen El Nino, allerdings mit etwas geringerem Ausmaß. Die Information dieses Phänomens scheint also in irgendeiner Form in dem verwendeten Eingabemuster gespeichert zu sein. Zwar ist der vom Netz prädizierte Effekt wesentlich größer

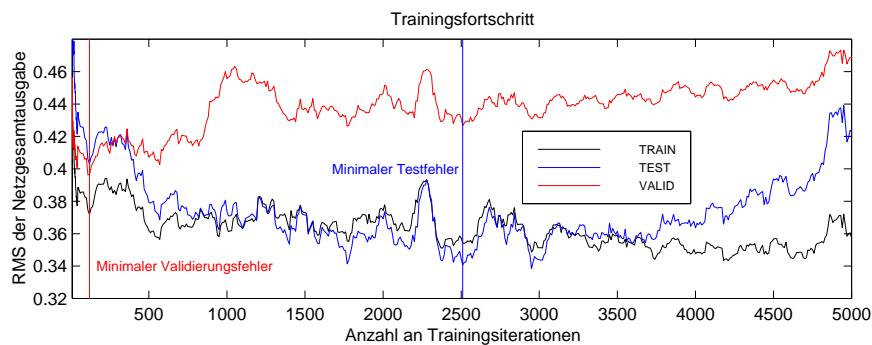


Abbildung 5.14: RMS-Netzausgabefehler während des Trainierens von ΔLOD

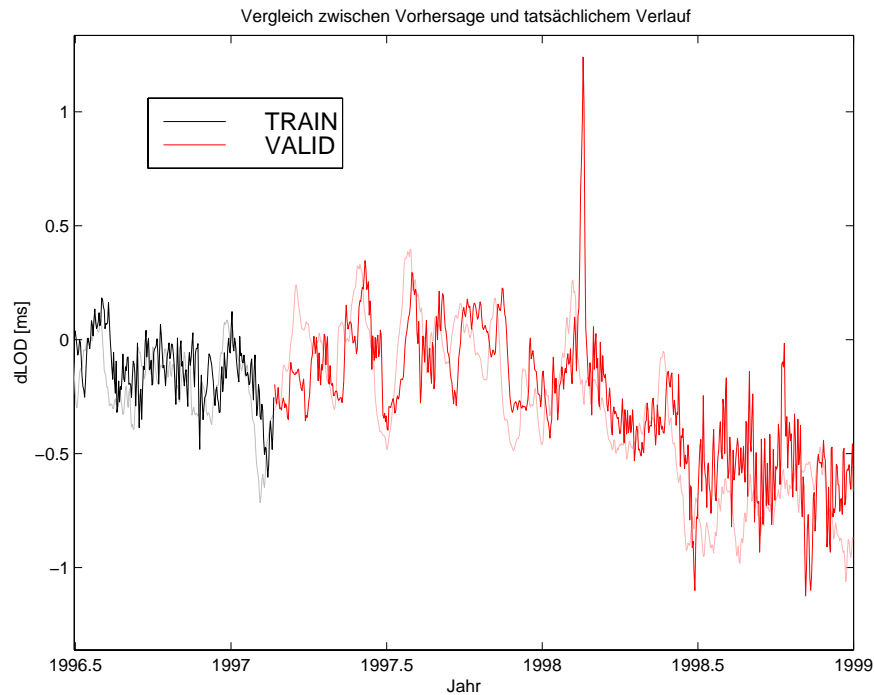


Abbildung 5.15: Vergleich zwischen Vorhersage des 10. Tages (dunkle Farbe) und tatsächlichem Verlauf (helle Farbe) von ΔLOD

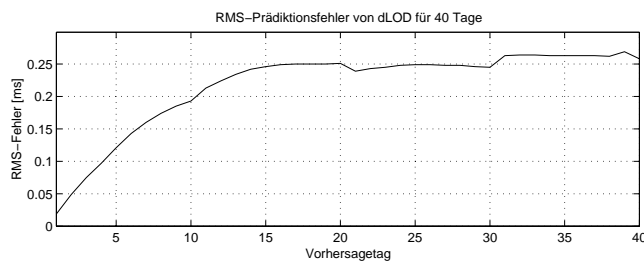


Abbildung 5.16: Prädiktionsfehler von ΔLOD

als der tatsächliche Ausschlag, dennoch ist es bemerkenswert, dass offensichtlich tatsächlich eine Abnormität vom Netz in diesem Zeitraum festgestellt wird.

Vorhersage der Tage 11 bis 40

Die Vorhersage der folgenden Tage bis einschließlich des 40. Tages wird – wie oben – in drei Teilbereiche mit je 10 Vorhersagetagen aufgespalten. Die resultierenden Prädiktionsfehler sind im Folgenden zusammengestellt:

Vorhersagetag	5	10	15	20	25	30	35	40
$RMS^{\Delta LOD}$ [ms]	0.121	0.193	0.246	0.251	0.249	0.245	0.263	0.258

Auch hier stellt sich heraus, dass für Vorhersagen, die weiter in die Zukunft reichen, weniger Trainingsiterationen durchlaufen werden müssen, um einen minimalen Netzausgabefehler des Testdatensatzes zu erreichen.

Wie in Abbildung 5.16 gut zu erkennen ist, steigt der Fehler zunächst fast linear an und konvergiert anscheinend mit zunehmendem Vorhersagezeitraum gegen einen konstanten Wert. Dieser liegt ungefähr bei 0.26 ms. Dieser Umstand ist wohl auf die sehr gute Reduktion der Datenreihe im Vorfeld zurückzuführen.

Die wannenförmige Vertiefung der Fehlerkurve zwischen dem 21. und 30. Vorhersagetag ist damit zu erklären, dass bei der Vorhersage dieses Zeitraumes die beiden Fehlerminima der Test- und Validierungsmuster sehr eng zusammenliegen. Deshalb liegt nach dem Trainingsende, das durch das Fehlerminimum der Testmuster bestimmt ist,

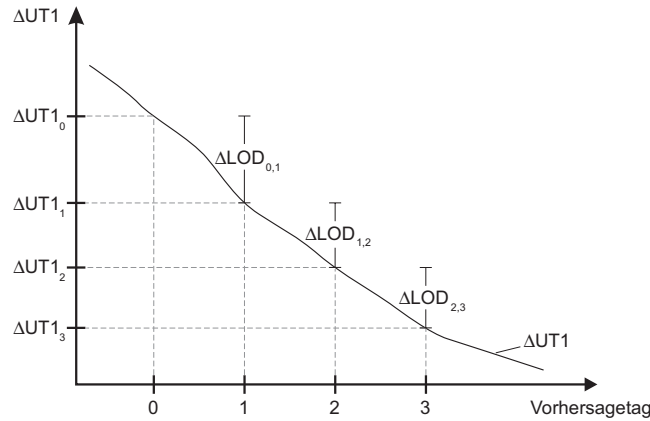


Abbildung 5.17: Zusammenhang zwischen $\Delta UT1$ und ΔLOD im diskreten Fall

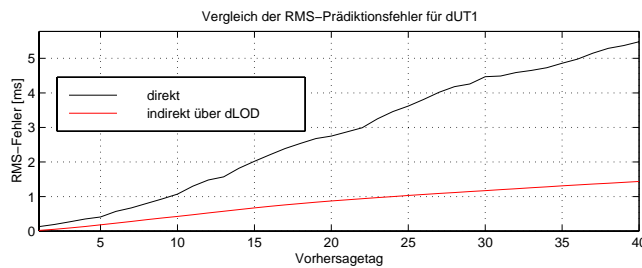


Abbildung 5.18: Vergleich der direkten Prädiktion von $\Delta UT1$ mit der Umrechnung aus ΔLOD

nahezu ein Fehlerminimum der Validierungsmuster und damit der Vorhersage vor. In den drei übrigen Zeiträumen der Vorhersage ist das Fehlerminimum der Validierungsmuster sehr viel früher erreicht als das der Testmuster, sodass der Prädiktionsfehler bereits wieder angestiegen ist. Dies zeigt deutlich, dass es in der Praxis durchaus zu Genauigkeitsschwankungen kommen kann.

Vergleich des Ergebnisses mit $\Delta UT1$

Um das Ergebnis mit dem der Vorhersage von $\Delta UT1$ vergleichen zu können, wenden wir das Fehlerfortpflanzungsgesetz (FFG) an. $\Delta UT1$ können wir als Integral von ΔLOD schreiben (siehe Kapitel 2.1.1). Im diskreten Fall bedeutet dies eine Summation der täglichen ΔLOD -Werte (vgl. Abbildung 5.17). Die Integrationskonstante $\Delta UT1_0$ wählen wir so, dass $\Delta UT1_n$ der Prädiktion des n . Tages entspricht:

$$\Delta UT1_n = \Delta UT1_0 - (\Delta LOD_{0,1} + \Delta LOD_{1,2} + \dots + \Delta LOD_{n-1,n}) \tag{5.5}$$

$\Delta UT1_0$ ist demnach der letzte bekannte Wert und ist somit fehlerfrei bezüglich der Vorhersage. Liegen, wie in unserem Fall, Genauigkeiten für $\Delta LOD_{i,i+1}$ vor, so ist nach Anwendung des Fehlerfortpflanzungsgesetzes die Genauigkeit von $\Delta UT1_n$ unter der Annahme unkorrelierter Werte für $\Delta LOD_{i,i+1}$

$$\sigma_{\Delta UT1_n}^2 = \sigma_{\Delta LOD_{0,1}}^2 + \sigma_{\Delta LOD_{1,2}}^2 + \dots + \sigma_{\Delta LOD_{n-1,n}}^2 \tag{5.6}$$

Nach der Umrechnung der Genauigkeiten ergeben sich folgende Werte für die Prädiktion von $\Delta UT1$:

Vorhersagetag	5	10	15	20	25	30	35	40
$RMS^{\Delta UT1}$ [ms] (direkt)	0.41	1.07	2.02	2.75	3.62	4.47	4.86	5.48
$RMS^{\Delta UT1}$ [ms] (über ΔLOD)	0.18	0.42	0.67	0.87	1.03	1.17	1.31	1.44

Das Ausmaß dieser scheinbaren Genauigkeitssteigerung gegenüber der direkten Vorhersage von $\Delta UT1$ verdeutlicht Abbildung 5.18, die beide Fehlerkurven gegenüberstellt.

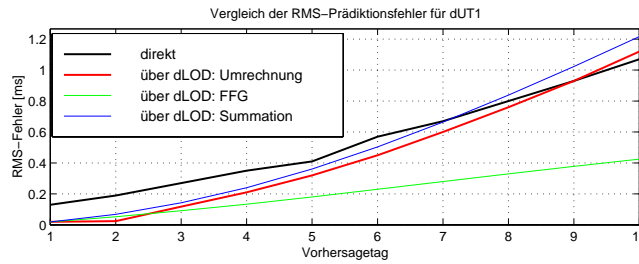


Abbildung 5.19: Fehlervergleich: $\Delta UT1$ und ΔLOD bis zum 10. Vorhersagetag

Da die abgeleiteten Fehler unrealistisch klein ausfallen, wird zur Überprüfung ein zweites Umrechnungsverfahren angewendet: Zunächst addieren wir die bekannten Einflüsse zurück auf die vorhergesagten residualen ΔLOD -Werte. Die Ergebnisse können dann über den funktionalen Zusammenhang aus Formel 5.5 in $\Delta UT1$ umgerechnet werden. Durch Gegenüberstellung der dadurch erhaltenen Werte mit den tatsächlichen $\Delta UT1$ -Werten aus C04 kann abermals für jeden Vorhersagetag ein RMS-Fehler errechnet werden. Bei der Anwendung der Formel 5.5 ist Vorsicht geboten: Die benötigten $\Delta LOD_{i,i+1}$ -Werte müssen durch arithmetische Mittelbildung von ΔLOD_i und ΔLOD_{i+1} berechnet werden. Das Ergebnis für die ersten 10 Vorhersagetage zeigt folgende Tabelle:

Vorhersagetag	1	2	3	4	5	6	7	8	9	10
$RMS^{\Delta UT1}$ [ms] (direkt)	0.13	0.19	0.27	0.35	0.41	0.57	0.67	0.80	0.93	1.07
$RMS^{\Delta UT1}$ [ms] (über ΔLOD : FFG)	0.02	0.05	0.09	0.13	0.18	0.23	0.28	0.33	0.37	0.42
$RMS^{\Delta UT1}$ [ms] (über ΔLOD : Umrechnung)	0.02	0.05	0.12	0.21	0.32	0.45	0.60	0.76	0.93	1.12

Zwar ergeben sich nun wesentlich höhere Werte als über das FFG, trotzdem ist die Qualität der ersten acht Vorhersagetage teilweise deutlich besser als bei der direkten Vorhersage. Die Fehlerberechnung über FFG bringt vermutlich deshalb zu optimistische Ergebnisse hervor, weil wir von unkorrelierten ΔLOD -Werten ausgehen. Tatsächlich sind benachbarte Werte wahrscheinlich sehr stark miteinander korreliert, da die Stützstellen sehr eng beieinanderliegen. Da uns jedoch die Kovarianzen nicht bekannt sind, kann diese Vermutung nicht durch Nachrechnen überprüft werden. Bei der Wertung der Ergebnisse darf dieser Umstand aber auf keinen Fall übersehen werden.

Interessant ist nun noch der Vergleich der Ergebnisse bis zum 40. Vorhersagetag mit einer simplen Summation der RMS-Fehler für die ΔLOD -Vorhersage:

Vorhersagetag	5	10	15	20	25	30	35	40
$RMS^{\Delta UT1}$ [ms] (direkt)	0.41	1.07	2.02	2.75	3.62	4.47	4.86	5.48
$RMS^{\Delta UT1}$ [ms] (über ΔLOD : FFG)	0.18	0.42	0.67	0.87	1.03	1.17	1.31	1.44
$RMS^{\Delta UT1}$ [ms] (über ΔLOD : Umrechnung)	0.32	1.12	1.99	3.04	4.10	5.18	6.33	7.24
$RMS^{\Delta UT1}$ [ms] (über ΔLOD : Fehlersummation)	0.36	1.22	2.38	3.62	4.85	6.09	7.40	8.72

Die Wirklichkeit (Umrechnung) liegt also zwischen den beiden Extrema FFG und Fehlersummation, wobei letztere den tatsächlichen Fehler wesentlich besser beschreibt. Die Abbildungen 5.19 und 5.20 zeigen den Fehlerverlauf der vier Fälle.

Die Vorhersage des kurzfristigen Verlaufes von $\Delta UT1$ bis zum 10. Tag gelingt über die indirekte Vorhersage von ΔLOD und anschließende Umrechnung deutlich besser. Für die Vorhersage der Tage 11 bis 40 ist allerdings die direkte Vorhersage vorzuziehen.

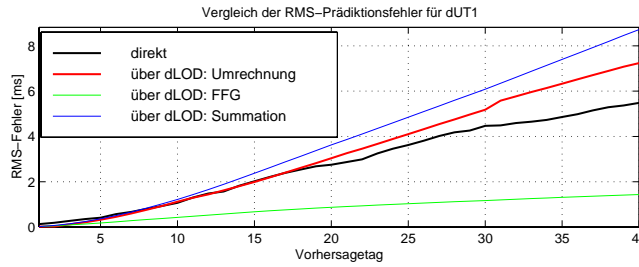


Abbildung 5.20: Fehlervergleich: $\Delta UT1$ und ΔLOD bis zum 40. Vorhersagetag

5.2 Mittelfristige Entwicklung

Bei der Prädiktion der Erdrotationsparameter innerhalb eines Jahres stellt sich die Frage, inwiefern sich eine Datenreduktion der saisonalen Schwankungen positiv auf das Ergebnis auswirkt. Die jährlichen und halbjährlichen Schwingungen aller Erdrotationsparameter und vor allem die Chandlersche Schwingungskomponente der Polbewegung unterliegen relativ starken Schwankungen. Eine Reduktion der Zeitreihen im Vorfeld ist jedoch nur dann sinnvoll, wenn auch innerhalb des Vorhersagezeitraums mit einer einigermaßen konstanten Fortsetzung gerechnet werden kann. Die Gezeiten hingegen sind für jeden Zeitpunkt der Zukunft exakt berechenbar und sind deshalb von dieser Problematik auszuschließen. Auf die Subtraktion des linearen Anteils ist aber in keinem Fall zu verzichten, da bei der gewählten Vorhersagestrategie eine Extrapolation nur im Wertebereich der Trainingsdaten verwertbare Ergebnisse liefert (vgl. Kapitel 3.2.2, Konfiguration 6).

Für die Vorhersage stellte sich folgende Mustereinteilung als günstig heraus:

Eingabemuster: [1 30 60 90 120 150 180 210 240 270 300 330 360 420 540 780 1260]

Ausgabemuster: [30 60 90 120 150 180 210 240 270 300 330 360]

Die verwendete Konfiguration entspricht im Übrigen derjenigen aus den vorangegangenen Kapiteln.

Um die Auswirkungen einer Datenreduktion für die mittelfristige Vorhersage gemäß obiger Überlegung nachzuvollziehen, sollen für jeden Erdrotationsparameter zwei verschiedene Reduktionsstufen untersucht werden:

1. Die Datenreihen werden lediglich um den Effekt der Gezeiten und den linearen Anteil reduziert. Saisonale Schwankungen und die Chandlersche Schwingung verbleiben im Datensatz.
2. Alle bekannten Effekte werden wie bisher vollständig reduziert.

Die Ergebnisse lassen sich am besten mit Hilfe der Fehlerkurven aus Abbildung 5.21 bewerten. Die mittelfristige Vorhersage der Erdrotationsparameter gelingt also in den meisten Fällen besser, wenn im Vorfeld alle bekannten Einflüsse reduziert worden sind. Der Vorteil einer Datenreduktion überwiegt also gegenüber dem Nachteil der Variabilität der saisonalen Einflüsse. Die numerischen Ergebnisse der Prädiktion mit vollständiger Reduktion sind in nachstehender Aufstellung zusammengefasst:

Vorhersagetag	30	60	90	120	150	180	210	240	270	300	330	360
RMS^{pol} [mas]	9.5	13.2	18.2	21.9	22.9	23.7	24.1	25.3	24.5	24.3	23.4	25.1
$RMS^{\Delta UT1}$ [ms]	7.4	11.1	14.9	17.0	16.2	15.3	15.0	17.3	19.4	20.7	21.5	22.9
$RMS^{\Delta LOD}$ [ms]	0.26	0.29	0.31	0.31	0.33	0.36	0.40	0.38	0.39	0.40	0.37	0.35

5.3 Verwendung der Zeitreihe SPACE98

Die SPACE98-Reihe von Gross (1999) beinhaltet Werte für die Erdrotationsparameter und deren Genauigkeiten innerhalb der Zeitspanne vom 28. September 1976 bis zum 16. Januar 1999. Die Daten sind wie bei der C04-Reihe im Tagesabstand gegeben. Abbildung 5.22 zeigt die Differenzen aller Parameter zwischen der C04- und der SPACE98-Reihe. Deutlich ist zu erkennen, dass die Unterschiede beider Reihen vor 1980 wesentlich größer sind

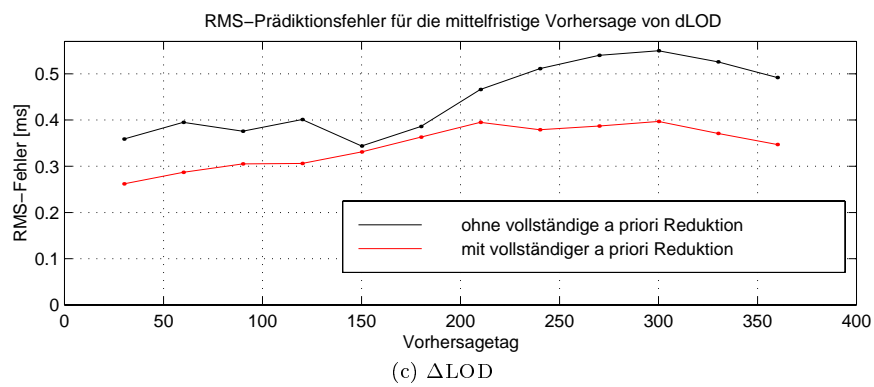
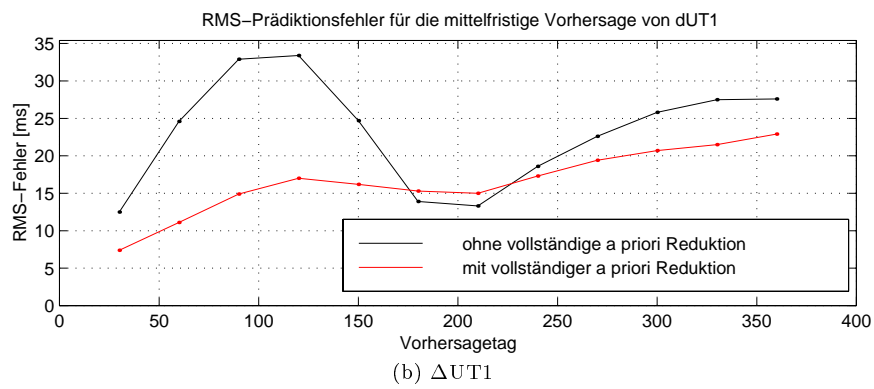
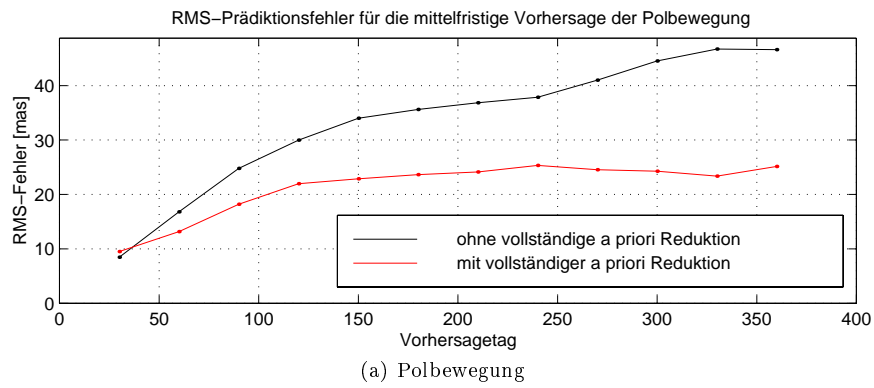


Abbildung 5.21: Vergleich der Vorhersageergebnisse bei Anwendung unterschiedlicher a priori Reduktionsstufen der Daten

Differenzen zwischen C04 und SPACE98

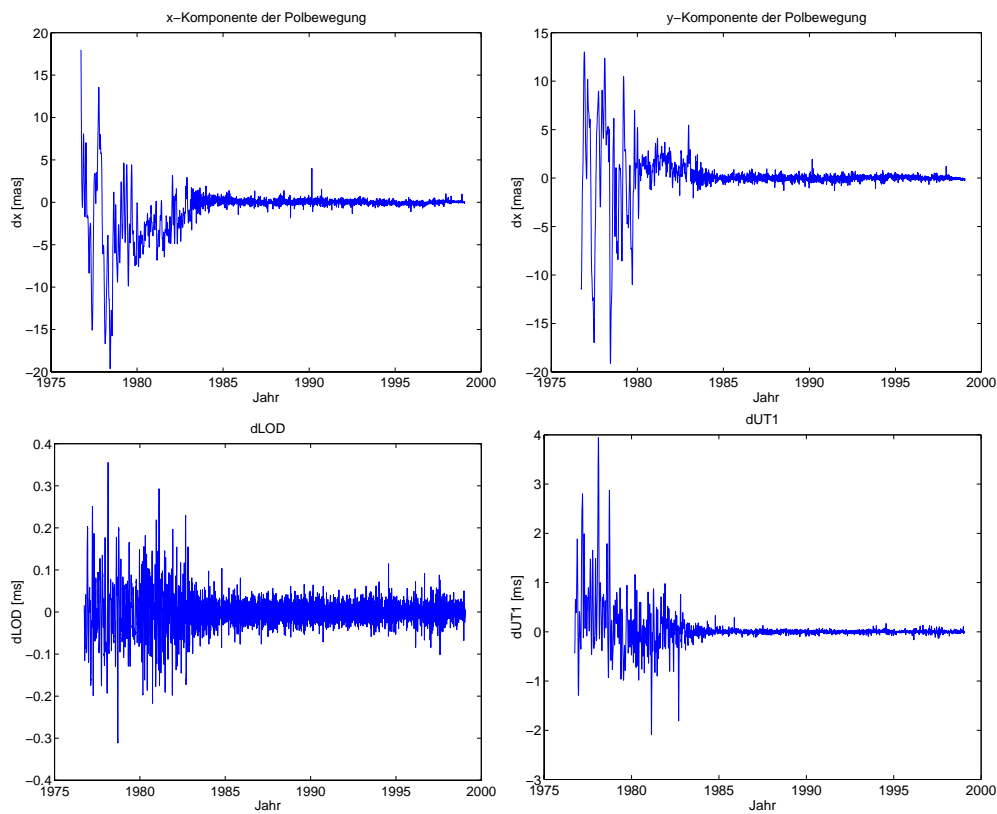


Abbildung 5.22: Differenzen C04 - SPACE98

als nach diesem Jahr. Im Jahre 1984 ist erneut eine Reduzierung der Differenzen festzustellen. Die Verringerung der Abweichungen ist auf die gestiegene Messgenauigkeit zurückzuführen. Es ist fraglich, ob die Verwendung der Zeitreihe SPACE98 im Gegensatz zur Zeitreihe C04 signifikante Genauigkeitsunterschiede bei der Prädiktion liefert, da beide Reihen für den größten Teil des für die Vorhersage verwendeten Zeitraumes vom 1.1.1980 bis zum 31.12.1998 nur unterhalb der Vorhersagegenauigkeit differieren. Eventuelle Unterschiede in den Ergebnissen der Vorhersage beruhen dann wohl auf den Diskrepanzen zwischen 1980 und 1984.

Die Prädiktion der Tage 1 bis 10 soll als Vergleichskriterium dienen, um zu entscheiden, ob die Auswahl der "richtigen" Datenreihe für das Ergebnis der Vorhersage eine Rolle spielt. Die Vergleichbarkeit der Ergebnisse wird dadurch gewährleistet, dass alle einstellbaren Parameter mit den Werten besetzt werden, die auch schon bei der Datenreihe C04 angewendet wurden. Die Musterauf- und einteilung werden ebenfalls unverändert übernommen. Selbstverständlich bleibt das verwendete Datenfenster (1980 bis 1998) ebenso bestehen. Die Ergebnisse sind so dargestellt, dass ein direkter Vergleich mit den Prädiktionen von C04 möglich ist.

Polbewegung

Die Vorhersage der Tage 1 bis 10 wird wieder aufgeteilt in eine gemeinsame Vorhersage der Komponenten für die Tage 1 bis 5 und eine getrennte Vorhersage für die Tage 6 bis 10. Die Ergebnisse der Reihen C04 und SPACE weichen nur geringfügig voneinander ab:

Vorhersagetag		1	2	3	4	5	6	7	8	9	10
RMS^x [mas]	SPACE	0.47	0.73	1.07	1.51	1.93	2.33	2.71	3.09	3.45	3.87
	C04	0.34	0.67	1.08	1.58	2.18	2.46	2.82	3.17	3.52	3.89
RMS^y [mas]	SPACE	0.26	0.51	0.80	1.08	1.35	1.60	1.82	2.02	2.22	2.41
	C04	0.23	0.46	0.68	0.95	1.28	1.67	1.87	2.06	2.25	2.44
RMS^{pol} [mas]	SPACE	0.38	0.63	0.94	1.31	1.67	1.99	2.31	2.61	2.90	3.22
	C04	0.29	0.57	0.90	1.30	1.79	2.10	2.39	2.67	2.95	3.25

$\Delta UT1$

Vorhersagetag		1	2	3	4	5	6	7	8	9	10
$RMS^{\Delta UT1}$ [ms]	SPACE	0.10	0.15	0.21	0.30	0.34	0.45	0.51	0.70	0.79	0.98
	C04	0.13	0.19	0.27	0.35	0.41	0.57	0.67	0.80	0.93	1.07

Die Ergebnisse der Vorhersage mit der SPACE-Reihe sind durchgängig besser als mit C04.

ΔLOD

Vorhersagetag		1	2	3	4	5	6	7	8	9	10
$RMS^{\Delta LOD}$ [ms·10 ⁻²]	SPACE	2.7	5.2	7.2	9.1	12.9	15.0	17.0	18.5	19.8	20.8
	C04	1.9	4.9	7.4	9.7	12.1	14.2	15.9	17.4	18.4	19.3

Ähnlich wie bei der Polbewegung sind auch hier nur geringfügige Unterschiede zu erkennen. Lediglich die Vorhersage des 1. Tages gelingt mit der SPACE-Reihe sowohl bei der Polbewegung als auch bei ΔLOD weniger gut.

Mittelfristige Vorhersage

Aufgrund der deutlich zu erkennenden Genauigkeitsunterschiede bei $\Delta UT1$ soll die Qualität einer weiterführenden Vorhersage bis zu einem Jahr auf der Grundlage der SPACE-Reihe untersucht werden. Aus Gründen der Vollständigkeit sind im Folgenden auch die Prädiktionsergebnisse der Polbewegung und die von ΔLOD aufgelistet:

Vorhersagetag		20	30	40	60	120	180	240	300	360
RMS^{pol} [mas]	SPACE	6.80	8.44	11.1	14.9	15.1	17.8	25.8	23.1	26.2
	C04	6.28	8.89	11.0	13.2	21.9	23.7	25.3	24.3	25.1
$RMS^{\Delta UT1}$ [ms]	SPACE	2.89	4.76	5.71	13.6	19.2	18.5	21.9	29.5	34.6
	C04	2.75	4.47	5.48	11.1	17.0	15.3	17.3	20.7	22.9
$RMS^{\Delta LOD}$ [ms]	SPACE	0.245	0.255	0.250	0.27	0.33	0.34	0.37	0.41	0.34
	C04	0.251	0.245	0.258	0.29	0.31	0.36	0.38	0.40	0.35

Für die Vorhersage der Tage 11 bis 40 von $\Delta UT1$ liefert die SPACE-Reihe jetzt sogar schlechtere Ergebnisse. Dieser Unterschied wird mit zunehmendem Vorhersagezeitraum noch größer. Die Ursache für die relativ schlechte Vorhersage von $\Delta UT1$ liegt darin, dass die Fehlerminima der Test- und Validierungsdaten wieder sehr weit auseinander liegen. Der tatsächliche Netzausgabefehler der Validierungsdaten ist somit wesentlich höher als sein Minimum. Würden wir das Training am Minimum der Validierungsdaten abbrechen, so hätte beispielsweise der 360. Vorhersagetag einen RMS-Fehler von nur 21.0 ms. Dies ist ein in der Praxis nicht zu vernachlässigendes Problem, da die Genauigkeitsschwankungen bei $\Delta UT1$ doch relativ hoch sind.

Für die beiden anderen Rotationsparameter sind dagegen keine systematischen Veränderungen feststellbar.

5.4 Anwendung: Vorhersage für das Jahr 2000

Um die gesammelten Erfahrungen in der Praxis zu testen, werden die Erdrotationsparameter ab dem 8.3.2000 für ein Jahr vorhergesagt. Für diesen Zeitraum existieren Prädiktionen des IERS für die Polbewegung und $\Delta UT1$ im Tagesabstand, die vom *Sub-Bureau for Rapid Service and Prediction* im Bulletin-A veröffentlicht werden und als Vergleichswerte dienen sollen.

Für die Vorhersage mit neuronalen Netzen verwenden wir die Erdrotationsparameter der C04-Reihe des IERS vom 1.1.1980 bis zum 7.3.2000. Aus diesen Daten werden jeweils Muster für die entsprechende Prädiktion generiert. Die ersten 10 Prozent dieser Muster werden wieder jeweils vom Training ausgespart und fungieren als Abbruchindikator für den Trainingsvorgang.

Für jeden Parameter existieren vier separate Vorhersagen im kurzfristigen Bereich bis zu 40 Tagen und eine im mittelfristigen Bereich bis zu 360 Tagen. Bei der Polbewegung werden zusätzlich die ersten 5 Tage getrennt

behandelt (siehe Kapitel 5.1.1). Alle Einzelvorhersagen werden dann aneinandergereiht und sind in folgender Tabelle zusammengefasst:

Vorher- sage- tag	Datum	Polbewegung		UT1 – UTC [s]	Δ LOD [ms]
		x [as]	y [as]		
1	8.3.2000	0.06994	0.35981	0.2967523	1.3652
2	9.3.2000	0.07065	0.35941	0.2950542	1.4015
3	10.3.2000	0.07016	0.35888	0.2933700	1.3674
4	11.3.2000	0.07137	0.35838	0.2915842	1.2802
5	12.3.2000	0.07198	0.35786	0.2899955	1.1670
6	13.3.2000	0.07291	0.35738	0.2885208	1.0759
7	14.3.2000	0.07378	0.35699	0.2870924	1.0169
8	15.3.2000	0.07504	0.35643	0.2856255	1.0305
9	16.3.2000	0.07517	0.35596	0.2841884	1.1104
10	17.3.2000	0.07636	0.35544	0.2827651	1.2532
11	18.3.2000	0.08307	0.34734	0.2633232	1.3837
12	19.3.2000	0.08367	0.34685	0.2613632	1.5194
13	20.3.2000	0.08428	0.34611	0.2593597	1.6153
14	21.3.2000	0.08488	0.34537	0.2573546	1.5683
15	22.3.2000	0.08520	0.34461	0.2551608	1.4956
16	23.3.2000	0.08591	0.34387	0.2533147	1.4076
17	24.3.2000	0.08632	0.34327	0.2513772	1.2404
18	25.3.2000	0.08700	0.34263	0.2498061	1.1100
19	26.3.2000	0.08715	0.34183	0.2482142	1.0061
20	27.3.2000	0.08803	0.34104	0.2467928	0.9100
21	28.3.2000	0.09690	0.33684	0.2307544	0.9144
22	29.3.2000	0.09718	0.33589	0.2291796	0.9168
23	30.3.2000	0.09756	0.33544	0.2278128	1.0200
24	31.3.2000	0.09853	0.33480	0.2263479	1.1203
25	1.4.2000	0.09877	0.33407	0.2246536	1.2498
26	2.4.2000	0.09896	0.33364	0.2228042	1.4626
27	3.4.2000	0.09945	0.33289	0.2209463	1.5915
28	4.4.2000	0.09988	0.33238	0.2188701	1.7071
29	5.4.2000	0.10029	0.33175	0.2166826	1.7879
30	6.4.2000	0.10075	0.33100	0.2145673	1.7632
31	7.4.2000	0.10248	0.32557	0.1931574	1.7405
32	8.4.2000	0.10269	0.32394	0.1910154	1.6114
33	9.4.2000	0.10313	0.32319	0.1890352	1.4259
34	10.4.2000	0.10319	0.32228	0.1871943	1.3179
35	11.4.2000	0.10317	0.32173	0.1848818	1.2747
36	12.4.2000	0.10338	0.32147	0.1837648	1.3339
37	13.4.2000	0.10360	0.32026	0.1820727	1.4344
38	14.4.2000	0.10387	0.31925	0.1802098	1.5625
39	15.4.2000	0.10401	0.31886	0.1781764	1.6802
40	16.4.2000	0.10442	0.31794	0.1760223	1.7381
60	6.5.2000	0.10513	0.29583	0.1141157	1.2801
90	5.6.2000	0.09760	0.27072	0.0590060	0.5208
120	5.7.2000	0.08030	0.25474	0.0074191	0.4003
150	4.8.2000	0.04709	0.23997	-0.0388843	0.5022
180	3.9.2000	0.01311	0.24626	-0.0988860	0.5709
210	3.10.2000	-0.02832	0.26720	-0.1685564	0.5814
240	2.11.2000	-0.05783	0.30525	-0.2508977	0.4375
270	2.12.2000	-0.08125	0.35293	-0.3313463	0.5597
300	1.1.2001	-0.07548	0.40299	-0.4022213	0.7024

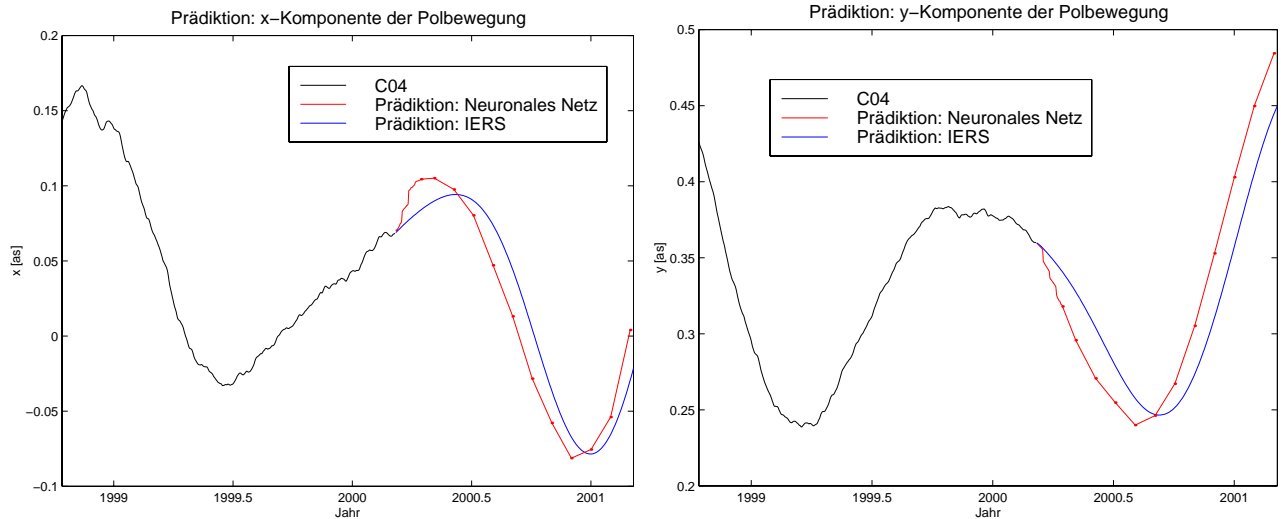


Abbildung 5.23: Prädiktion der Polbewegung

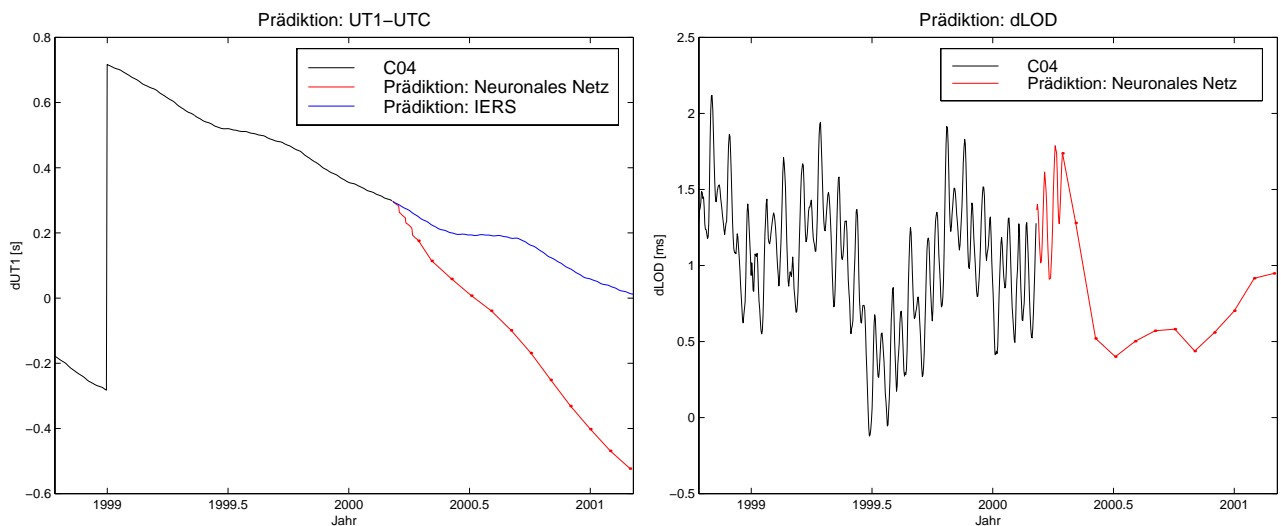


Abbildung 5.24: Prädiktion von UT1 – UTC und Δ LOD

Vorher- sage- tag	Datum	Polbewegung		UT1 – UTC [s]	Δ LOD [ms]
		x [as]	y [as]		
330	31.1.2001	-0.05390	0.44981	-0.4687717	0.9161
360	2.3.2001	0.00408	0.48448	-0.5229839	0.9483

Zu beachten ist, dass bei der Darstellung von UT1 – UTC alle bisher eingeführten Schaltsekunden bereits berücksichtigt sind. Zukünftige Schaltsekunden können selbstverständlich nicht mit einkalkuliert werden.

Eine etwas anschaulichere Darstellung der Prädiktionsdaten ist in den Abbildungen 5.23 und 5.24 gegeben. Dort sind dem tatsächlichen Verlauf (C04) der letzten Tage sowohl die Vorhersageergebnisse mit neuronalen Netzen als auch die Ergebnisse des IERS gegenübergestellt. Leider sind im Bulletin-A keine Prädiktionswerte für Δ LOD enthalten, weshalb auf einen Vergleich verzichtet werden muss.

Die Ergebnisse dieser Arbeit sind in den Graphiken durch rote Linien gekennzeichnet. Diese sind im Bereich der mittelfristigen Vorhersage zusätzlich mit Punkten markiert, die auf die Äquidistanz von 30 Tagen hinweisen sollen. Zwischen den einzelnen Punkten liegt folglich keine Information über den Verlauf der Kurven vor. Während dies bei der Polbewegung und UT1 – UTC keine Nachteile mit sich bringt, geht bei der Darstellung von Δ LOD Information verloren, da die Auflösung weit unter der maximal auftretenden Frequenz liegt.

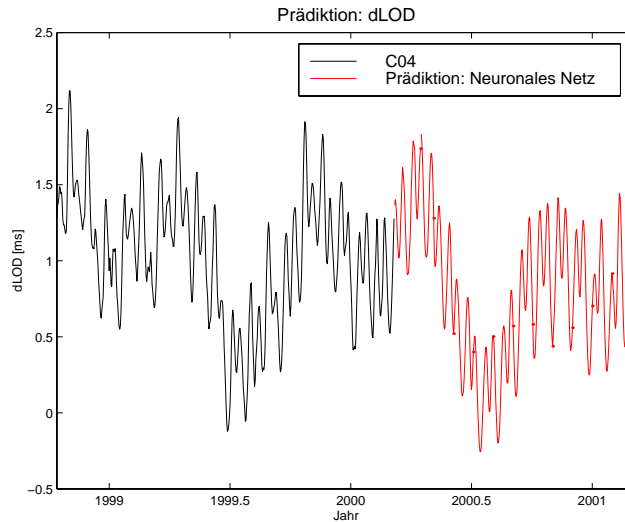


Abbildung 5.25: Prädiktion von ΔLOD nach Reduzierung der Stützpunktabstände von 30 Tagen auf 1 Tag bei der mittelfristigen Vorhersage durch Verschieben des Eingabemusters in Richtung Vergangenheit

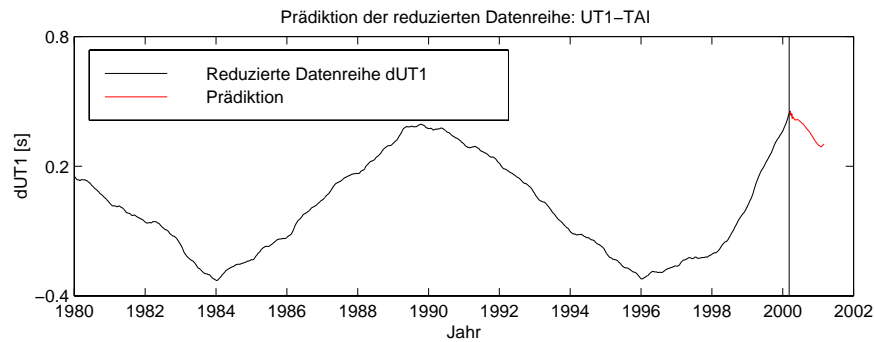


Abbildung 5.26: Reduzierte Datenreihe $\Delta UT1$ (Test- und Trainingsdaten) und Prädiktionswerte der neuronalen Netze

Obwohl wir die Vorhersagezeitpunkte nach der Wahl der Mustereinteilung nicht mehr verändern können, ist es mit Hilfe eines kleinen Tricks dennoch möglich, auch zwischen den Stützstellen Information über den Kurvenverlauf zu gewinnen: Ausgehend von einer mittelfristigen Vorhersage, die uns die Werte in 30, 60, 90...360 Tagen liefert, verschieben wir unser Eingabefenster um einen Tag in die Vergangenheit, sodass der letzte bekannte Wert für ΔLOD nicht mehr berücksichtigt wird. Legen wir dieses Eingabemuster an unser bereits trainiertes neuronales Netz, so erhalten wir Netzausgaben für die Vorhersagetage 29, 59, 89...359. Durch 29-maliges Verschieben können wir so die Lücken zwischen den ursprünglichen Vorhersagetagen schließen. Das Ergebnis ist in Abbildung 5.25 zu sehen. Zum Vergleich sind auch die alten Stützpunkte aus Abbildung 5.24 mit eingezeichnet.

Die Vorhersage der Polbewegung mit neuronalen Netzen weist einen zeitlichen Versatz zur IERS-Vorhersage auf (vgl. Abbildung 5.23). Der etwas "stufige" Verlauf innerhalb der ersten 40 Tage ist auf die Aufspaltung der kurzfristigen Vorhersage in 4 Abschnitte zurückzuführen, wobei zwischen dem letzten Ausgabeneuron eines Abschnittes und dem ersten des folgenden Abschnittes Unstetigkeiten auftreten. Ansonsten zeigen beide Vorhersagemethoden einen ähnlichen Verlauf.

Anders gestaltet sich der Vergleich bei $UT1 - UTC$ (vgl. Abbildung 5.24). Die Unstetigkeitsstelle Anfang des Jahres 1999 ist auf das Einfügen einer Schaltsekunde am 1.1.1999 um 0 Uhr zurückzuführen und ist für unsere Auswertung nicht weiter von Bedeutung. Bei der Vorhersage selbst ist allerdings ein deutlicher Knick zu erkennen, wenn wir von der bisherigen Entwicklung ausgehen. Hier scheint die Vorhersage des IERS einem wesentlich plausibleren Verlauf zu folgen. Bei der Ergründung dieser plötzlichen Steigungsänderung ist es sinnvoll, die Betrachtung auf die um die bekannten Einflüsse reduzierte Datenreihe zu beschränken. Sowohl der Verlauf der Test- und Trainingsdaten als auch die ausgegebenen Vorhersagewerte des neuronalen Netzes sind in Abbildung 5.26 aufgetragen. In dieser Graphik ist das Motiv des neuronalen Netzes für die "Unstetigkeit" in der ersten Ableitung zu erkennen: Vergleichen wir den Verlauf der letzten Jahre mit dem vor 1990, so sind durchaus Ähnlichkeiten feststellbar. Da

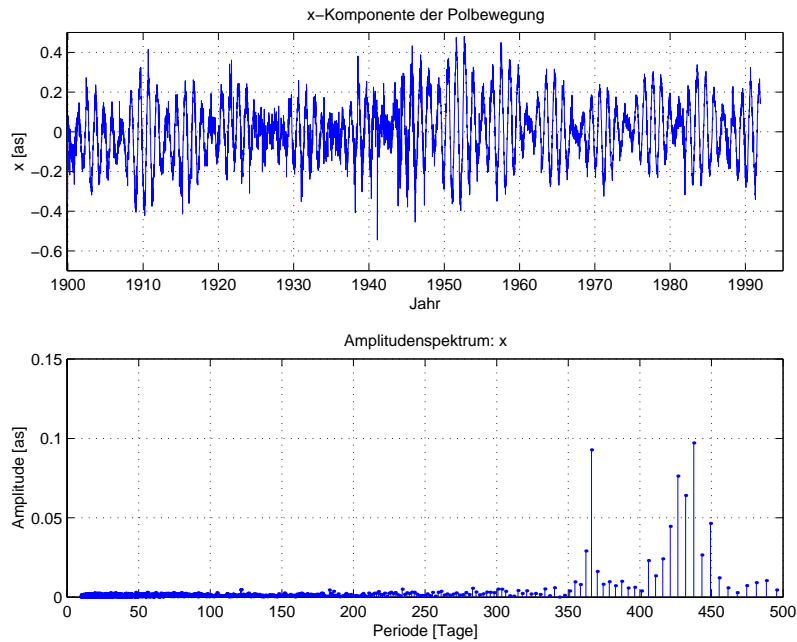


Abbildung 5.27: x-Komponente der Polbewegung aus der Vondrák-Reihe

die letzten bekannten Funktionswerte dazu noch ähnlich hoch sind wie die kurz vor 1990, ist es nun nur noch wenig verwunderlich, dass auch die Vorhersage dem Kurvenfortlauf ab 1990 nachempfunden ist und die Steigung ihr Vorzeichen wechselt. Ob sich allerdings auch die Realität an die Vorgabe der neuronalen Netze hält, bleibt abzuwarten.

5.5 Anwendung: Langfristige Entwicklung der Polbewegung

Zur Vorhersage der langfristigen Entwicklung der Polbewegungskomponenten verwenden wir die Zeitreihe von Vondrák, die aus astronomischen Beobachtungen ermittelt wurde. Als Grundlage für ihre Berechnung dienten die aus der HIPPARCOS-Mission ermittelten Sternkoordinaten. Die Polreihe enthält Werte für die Polbewegung seit dem 27.9.1899 bis zum 29.12.1991 mit einer Äquidistanz von 5 Tagen.

In den Abbildungen 5.27 und 5.28 sind neben den zeitlichen Verläufen auch die Spektren der Polbewegungskomponenten dargestellt. Der unregelmäßigere Verlauf der älteren Daten bis ca. 1950 aufgrund der damals geringeren Genauigkeit ist gut zu erkennen. Die starke Variation der Chandlerschen Periode wirkt sich im Spektrum durch eine Verbreiterung des Chandler-Peaks auf die umliegenden Perioden aus. Da nun die langfristige Vorhersage der Chandlerschen Schwingung und auch die der jährlichen Schwingung Gegenstand unserer Vorhersage sein soll, dürfen sie im Vorfeld nicht reduziert werden. Lediglich auf eine Reduktion des linearen Anteils kann nicht verzichtet werden, da ansonsten nicht sichergestellt wäre, dass der Wertebereich der Vorhersage Teilmenge des Wertebereichs der Trainingsdaten ist, was aber für die sinnvolle Anwendbarkeit von Vorhersagestrategie 3 unbedingt erforderlich ist.

Die langfristige Entwicklung der Polbewegung wird mit einem Vorhersagezeitraum von bis zu 40 Jahren abgedeckt. Eine weiter in die Zukunft reichende Prädiktion ist aufgrund des begrenzten Datenfensters der Vondrák Reihe (ca. 92 Jahre) nicht sinnvoll. Bedenkt man, dass bei einer Vorhersage von bis zu 40 Jahren das Eingabemuster sinnvollerweise die letzten ca. 40 Jahre überdecken sollte, so bleibt für den Verschiebungsspielraum dieses Musterpaars nur noch eine Zeitspanne von ungefähr 12 Jahren. Dies bedeutet, dass die Anzahl generierbarer Muster stark verkleinert wird. Eine weitere Dezimierung nach Abzug der Testmuster lässt darüber hinaus eine Ausdehnung des Vorhersagezeitraumes nicht sinnvoll erscheinen.

Den Vorgang der langfristigen Vorhersage können wir in drei Schritte gliedern:

1. Testen verschiedener Konfigurationen
2. Trainieren des Netzes

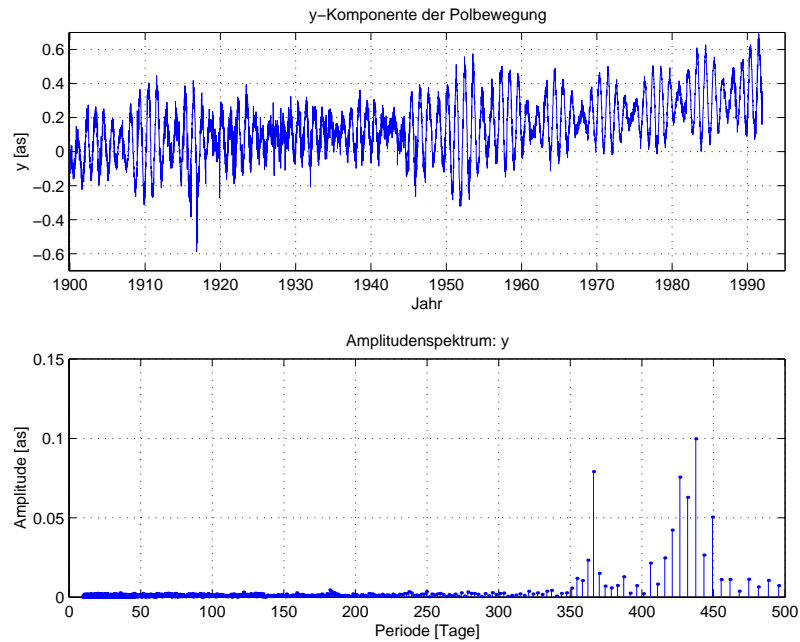


Abbildung 5.28: y-Komponente der Polbewegung aus der Vondrák-Reihe

3. Vorhersage

Im ersten Schritt werden - wie gewohnt - verschiedene Netztopologien und Mustereinteilungen getestet und bewertet. Anschließend wird die Konfiguration mit dem besten Ergebnis für die eigentliche Vorhersage verwendet. Der genaue Ablauf wird im Folgenden beschrieben.

5.5.1 Testen verschiedener Konfigurationen

Als Kriterien für die Qualität der getesteten Konfigurationen dienen die Netzausgabefehler der Musterdatensätze und die Anzahl der benötigten Trainingsiterationen. 90 Prozent aller generierbaren Muster werden wieder zum Trainieren und Testen genutzt, wovon wiederum die ersten 10 Prozent dem Testdatensatz zugeschrieben werden. Das Training wird beim Minimum des Netzausgabefehlers der Testdaten abgebrochen und das Ergebnis analysiert. Das Ausgabemuster beinhaltet die vorherzusagenden Tage im relativen Abstand von 5 Jahren:

Ausgabemuster in Jahren: [5 10 15 20 25 30 35 40]

In nachfolgender Übersicht sind alle getesteten Eingabemuster inklusive der RMS-Fehler des Ausgabeneurons, das den Vorhersagezeitpunkt in 40 Jahren repräsentiert, und der Anzahl an Iterationen zusammengestellt:

Eingabemuster [Jahre]	RMS_{TRAIN}^x [mas]	RMS_{TEST}^x [mas]	RMS_{VALID}^x [mas]	Iterationen
[0 1 2 3 4 5]	78.7	77.1	75.6	340
[0 1 2 3 4 5 6 7 8 9 10]	71.8	78.3	69.4	490
[0 5 10 15 20 25 30 35 40]	68.1	72.3	59.3	3100
[0 10 20 30 40]	69.8	73.4	72.1	450
[0 1 2 3 4 5 6 7 11 19 35]	67.7	72.1	54.9	1270

Die Null an erster Stelle der Eingabemuster soll darauf hindeuten, dass der erste Wert des Eingabemusters mit der letzten bekannten Beobachtung besetzt wird. Das in obiger Tabelle an letzter Stelle stehende Eingabemuster liefert die besten Ergebnisse. Die etwas höhere Iterationenzahl ist bei einer Trainingszeit von nur 15 Minuten zu vernachlässigen. Die Strategie, den relativen Abstand der Eingabewerte in Schritten von 2er-Potenzen zu erhöhen, bewährt sich auch hier. Auch die Information der letzten Jahre scheint wichtig zu sein.

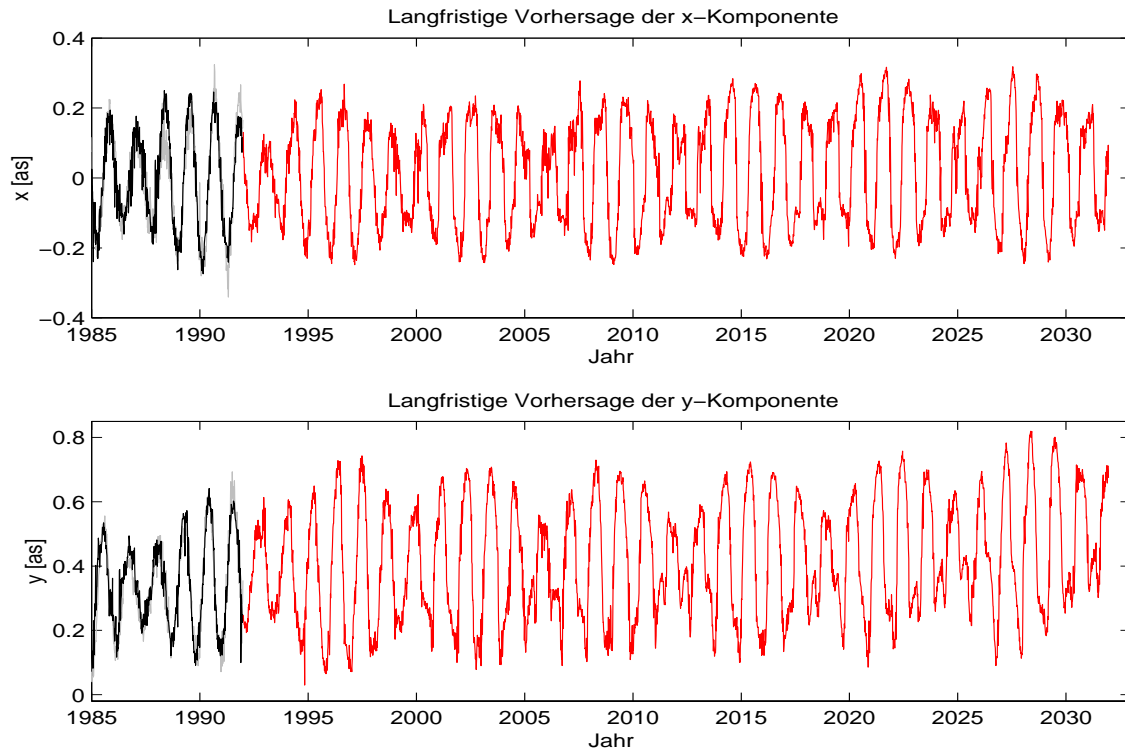


Abbildung 5.29: Ergebnis der Vorhersage mit einem Prädiktionsintervall von 40 Jahren: Der tatsächliche Verlauf ist grau, die Netzausgabe schwarz und die Vorhersage rot dargestellt

Ausgehend von der untersten Konfiguration werden nun verschiedene Topologien³ getestet:

Topologie	RMS_{TRAIN}^x [mas]	RMS_{TEST}^x [mas]	RMS_{VALID}^x [mas]	Iterationen
10-8-8-8	69.4	72.4	78.5	3880
10-8-4-8	67.7	72.1	54.9	1270
10-4-4-8	73.4	71.5	74.5	1160
10-4-2-8	67.2	71.7	52.9	2620
10-2-2-8	78.1	88.2	81.7	580

Das neuronale Netz mit 6 versteckten Neuronen bringt die besten Ergebnisse hervor.

5.5.2 Trainieren des Netzes

Zum eigentlichen Trainieren des oben gefundenen Netzes werden die Muster nur noch in Trainings- und Testmuster unterteilt. Die ersten 10 Prozent entsprechen den Testmustern, die letzten 90 Prozent den Trainingsmustern. Die Verbindungsgewichte zum Zeitpunkt des Fehlerminimums der Testmuster werden gespeichert.

5.5.3 Vorhersage

Für die Vorhersage wird an das trainierte Netz ein Musterdatensatz angelegt, der nur aus Eingabewerten besteht. Dieser knüpft an das letzte Musterpaar des Trainingsdatensatzes an und wird solange verschoben, bis das erste Eingabeneuron (entspricht dem Wert vor 5 Tagen) auf den Tag der letzten Beobachtung zu liegen kommt. Die Ausgaben des Netzes werden gespeichert und ausgewertet. Die Ergebnisse der Vorhersage des letzten Ausgabeneurons, d.h. der Prädiktion 40 Jahre in die Zukunft, ist in Abbildung 5.29 zu sehen.

³Alle Topologien basieren auf feedforward-Netzen und beinhalten Shortcut-Verknüpfungen.

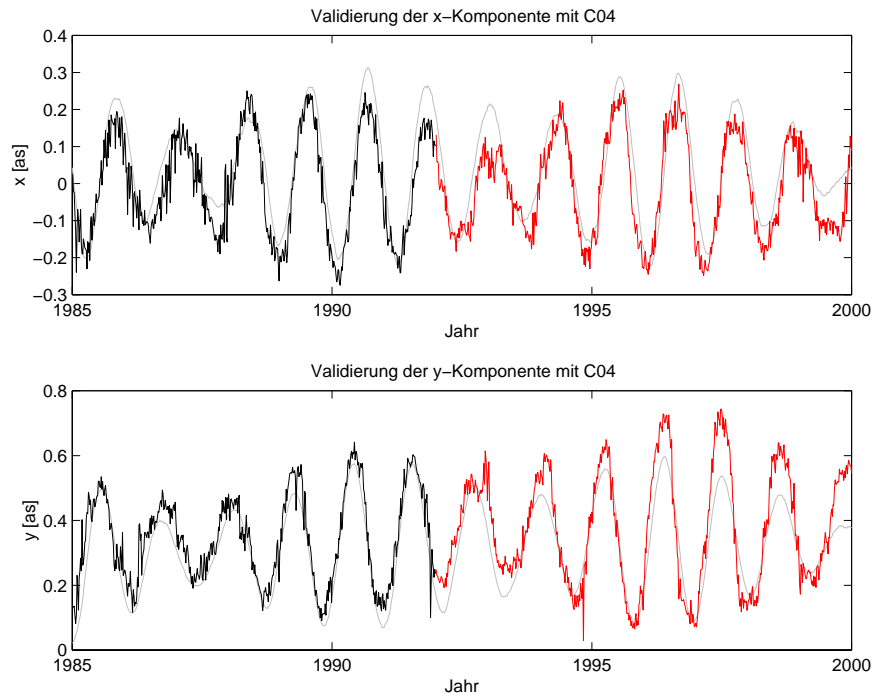


Abbildung 5.30: Validierung der langfristigen Vorhersage für 40 Jahre (schwarz und rot) mit der C04-Reihe des IERS (hellgrau) für den Zeitraum 1985 bis 2000

Die Trainingsdaten, deren Netzausgabe sich bis zum 29.12.1991 ausdehnt, werden erstaunlich gut repräsentiert, wenn wir uns die Länge des Vorhersageintervalls vor Augen halten. Auch die Vorhersage ab dem 30.12.1991 selbst folgt einem plausiblen Verlauf.

Zur Bewertung der Vorhersagegüte verwenden wir die Daten der C04-Reihe des IERS bis zum 31.12.1999. Die Abbildung 5.30 stellt einen Ausschnitt aus Abbildung 5.29 dar. Anstelle des tatsächlichen Verlaufes der Vondrák-Reihe ist nun allerdings die C04-Reihe gegenübergestellt (hellgrau). Beeindruckend ist die gute Übereinstimmung von Vorhersage und Wirklichkeit in Bezug auf Phase und Frequenz der Polbewegungskomponenten. Die Prädiktion der Amplitude gelingt dagegen weniger gut.

Kapitel 6

Weitere Vorhersagemethoden und Vergleich

In diesem Kapitel werden zunächst einige der wichtigsten Vorhersagemethoden und deren Ergebnisse vorgestellt. Anschließend wird das in dieser Arbeit angewendete Verfahren eingeordnet und das erzielte Ergebnis durch Vergleich mit den existierenden Methoden bewertet.

6.1 Vorgehen und Qualität weiterer Vorhersagemethoden

6.1.1 Vorhersagemethode nach Zhu (1981, 1982)

Diese Veröffentlichung beinhaltet Verfahren zur Vorhersage der Polbewegung und $\Delta UT1$.

Polbewegung (Zhu, 1981; Zhu 1982)

Das zugrunde liegende Modell für die Prädiktion der Polbewegung geht von den folgenden beiden Gleichungen aus:

$$x = x_0 + K_x(t - t_0) + b_x \sin(2\pi(t - t_0)) + c_x \cos(2\pi(t - t_0)) + B_C \sin \frac{2\pi(t - t_0)}{P_C} + C_C \cos \frac{2\pi(t - t_0)}{P_C} \quad (6.1)$$

$$y = y_0 + K_y(t - t_0) + b_y \cos(2\pi(t - t_0)) - c_y \sin(2\pi(t - t_0)) + B_C \cos \frac{2\pi(t - t_0)}{P_C} - C_C \sin \frac{2\pi(t - t_0)}{P_C} \quad (6.2)$$

Dabei werden die unbekannt Parameter K_x , b_x , c_x , K_y , b_y , c_y , B_C und C_C aus den Daten der letzten sechs oder sieben Jahre innerhalb einer Kleinsten-Quadrate-Ausgleichung geschätzt. Dieses Modell impliziert die Annahmen einer konstanten Periode P_C und einer kreisförmigen Bewegung der Chandlerschen Schwingung. Die halbjährliche Schwingung wird vernachlässigt. Berücksichtigt werden linearer Anteil und eine elliptische jährliche Schwingung mit konstanter Periode. Auf Grundlage der geschätzten Parameter werden die x- und y-Komponente für ein bis zwei Jahre extrapoliert. Die Prädiktionsfehler für verschiedene Vorhersagezeiträume zeigt folgende Tabelle:

Vorhersagetag	1	10	15	18.25	20	30	36.5	40	60
RMS^{pol} [mas]	2.3	4.5	6.6	7.1	8.7	12.3	13.3	15.5	20.9

Erdrotation (Zhu, 1981)

Zunächst werden die Parameter der jährlichen und halbjährlichen Schwingung und des linearen Anteils geschätzt. Dabei werden die Perioden des jährlichen und halbjährlichen Terms konstant gehalten:

$$UT1 - TAI = a_0 + a_1(t - t_0) + b \sin(2\pi(t - t_0)) + c \cos(2\pi(t - t_0)) + d \sin(4\pi(t - t_0)) + e \cos(4\pi(t - t_0)) \quad (6.3)$$

Dieses Modell wird anschließend von der Datenreihe subtrahiert. Die Residuen werden dann mit Hilfe einer linearen Filtermethode präzidiert und schlussendlich wieder zum Modell addiert. Das Ergebnis ist lediglich in Form einer Gegenüberstellung der Fehler der besten (+) und schlechtesten (-) Vorhersage gegeben:

Vorhersagetag	5	10	15	20	25	30	35	40
Fehler ⁺ [ms]	0.15	0.19	0.35	0.28	0.23	0.08	0.04	0.44
Fehler ⁻ [ms]	0.46	1.07	1.82	2.75	3.85	5.09	6.36	7.64

6.1.2 Vorhersagemethode nach Chao (1984)

Die Veröffentlichung von Chao (1984) beschreibt ein Verfahren zur Vorhersage der Polbewegung. Im Unterschied zu Zhu (1981) und Zhu (1982) werden die Perioden der jährlichen Schwingung und der Chandlerschen Schwingung nicht als konstant angenommen, sondern innerhalb der Ausgleichung mitgeschätzt. Ansonsten ähneln sich die Ansätze so stark, dass auf eine ausführlichere Beschreibung verzichtet werden kann.

Leider beziehen sich die von Chao angegebenen Fehlerwerte nicht auf einen individuellen Vorhersagetag. Zur Bestimmung der Genauigkeit einer Vorhersage bis zum Tage M wird vielmehr jeweils ein RMS-Wert aus den Fehlern der ersten M Tage errechnet. Als Qualitätsindikator dient dann der Mittelwert aller RMS-Fehler aus mehreren Vorhersagen mit konstanter Ausdehnung M. Aus diesen Fehlerwerten lässt sich also nicht auf den Prädiktionsfehler eines gewissen Tages schließen:

Vorhersagezeitspanne [Tage]	25	50	100	180	365
$\overline{RMS^x}$ [mas]	15.1	16.9	19.6	22.6	23.6
$\overline{RMS^y}$ [mas]	12.8	13.9	15.8	17.5	19.5
$\overline{RMS^{pol}}$ [mas]	14.1	15.5	17.8	20.2	21.6

Die errechneten Fehler sind kleiner als nach der Berechnung mit der herkömmlichen Methode, da auch Vorhersagen mit kürzerem Vorhersagezeitraum - und damit kleineren Fehlern - einfließen.

6.1.3 Vorhersagemethode nach McCarthy und Luzum (1990)

Die hier vorgestellten Vorhersagestrategien für die Polbewegung und UT1 – UTC werden vom *IERS Sub-Bureau For Rapid Service And Predictions* angewendet (IERS 1999).

Polbewegung

Jeweils die letzten beiden Jahre der Polbewegung werden dazu verwendet, wiederum die Parameter einer jährlichen elliptischen Schwingung, einer kreisförmigen Chandlerschen Schwingung und des linearen Anteils mit der Methode der kleinsten Quadrate zu schätzen. Die Differenzen zwischen den letzten Beobachtungen der Komponenten und den entsprechenden Werten aus den ausgeglichenen Kurven werden mit Hilfe einer linearen Gewichtsfunktion

$$\left(1 - \frac{T}{190}\right), \text{ mit } T < 190 \text{ Tage} \tag{6.4}$$

über die nächsten 190 extrapolierten Tage aufgeteilt. Dadurch wird ein stetiger Übergang zwischen den letzten beobachteten und den vorhergesagten Werten geschaffen. Ein Auszug aus der Ergebnistabelle ist im Folgenden verzeichnet:

Vorhersagetag	1	10	20	30	40	60	90	120	150	210
RMS^x [mas]	0.59	4.98	8.39	11.18	13.65	17.67	21.11	22.29	22.71	23.66
RMS^y [mas]	0.34	3.32	6.29	9.03	11.62	16.26	21.94	25.85	28.13	28.92
RMS^{pol} [mas]	0.48	4.23	7.41	10.16	12.68	16.98	21.53	24.14	25.56	26.42

Erdrotation

Für die Vorhersage von UT1 – UTC werden zunächst Schaltsekunden, Erdzeiten und saisonale Schwankungen modelliert und von der Datenreihe abgezogen. Die Vorhersagemethode selbst ist recht simpel: Um eine Prädiktion für n Tage in die Zukunft vorzunehmen, wird die Differenz zwischen dem letzten beobachteten Wert und dem Wert, der n Tage in der Vergangenheit liegt, gebildet. Diese Differenz wird anschließend wiederum zum letzten beobachteten Wert addiert:

$$\Delta UT1_n = 2\Delta UT1_0 - \Delta UT1_{-n} \tag{6.5}$$

Die Werte der Vergangenheit werden also lediglich am letzten beobachteten Wert (punkt-)gespiegelt.

Eine Verbesserung der Ergebnisse kann durch die Anwendung einer Gaußschen Glättung erzielt werden. Die Datenreihe wird im Vorfeld abhängig vom Vorhersagezeitraum unterschiedlich stark geglättet. Eine variable Glättungskonstante als Funktion des Vorhersagetages sorgt für eine schwache Glättung zur Vorhersage der kurzfristigen Schwankungen und für eine starke Glättung zur Vorhersage der langfristigen Entwicklung. Das Ergebnis zeigt folgende Tabelle:

Vorhersagetag	1	10	20	30	40	60	90	120	150	210
RMS^{UT1} [ms]	0.03	1.53	3.65	5.48	7.08	8.53	14.85	19.68	24.60	37.48

6.1.4 Vorhersagemethode nach Malkin und Skurikhina (1996)

Die hier beschriebene Vorgehensweise wird vom IAA EOP Service zur Vorhersage der Erdrotationsparameter genutzt.

Polbewegung

Zur Vorhersage der Polbewegung kommen hier zwei unterschiedliche Methoden zum Einsatz:

- **Deterministische Methode:**
 Sie beruht auf der Modellierung bekannter Komponenten unter Verwendung der Daten der letzten 1000 Tage und anschließender Extrapolation. Es werden jährliche und halbjährliche Schwingung, die Chandlersche Schwingungen und der lineare Anteil berücksichtigt. Um die Vorhersage des ersten Tages mit dem letzten beobachteten Wert in stetige Fortsetzung zu bringen, wird wie bei McCarthy und Luzum (1990) eine lineare Gewichtsfunktion verwendet. Anstatt der Übergangsdauer von 190 Tagen werden hier allerdings mit einem Gewichtsparameter von 180 Tagen bessere Ergebnisse erzielt. Diese Methode wird für Vorhersageintervalle genutzt, die 90 Tage überschreiten.
- **Stochastische Methode:**
 Auch hier werden im Vorfeld die oben genannten bekannten Komponenten modelliert. Die Datenreihe wird dann um diese reduziert. Die Residuen werden mit Hilfe der sog. ARIMA-Methode (Auto-Regressive Integrated Moving Average) unter Verwendung einer Maximum-Likelihood-Klassifikation zur Parameteranpassung vorhergesagt. Dieses Vorgehen wird zur Vorhersage der ersten 90 Tage genutzt.

Die Ergebnisse beider Methoden werden abschließend durch Offsetvariation und Glättung der Unstetigkeit an der Übergangsstelle zu einem Vorhersageergebnis verschmolzen. Drei unterschiedliche Intervalle werden zur Berechnung der Prädiktionsfehler herangezogen. Folgende Tabelle enthält die Mittelwerte der drei Intervalle:

Vorhersagetag	10	20	30	40	60	90	120	150	180
RMS^x [mas]	3.9	6.8	9.1	10.8	13.7	17.0	17.4	17.5	17.9
RMS^y [mas]	2.9	5.6	8.2	10.5	14.8	19.4	24.5	28.1	30.3
RMS^{pol} [mas]	3.4	6.2	8.7	10.7	14.3	18.2	21.2	23.4	24.9

Erdrotation

- Die ARIMA-Methode wird dazu verwendet, Vorhersagen für die nächsten 15 Tage zu treffen. Dazu werden vorweg linearer Anteil, jährliche und halbjährliche Schwingung und der Einfluss der Gezeiten von den Daten subtrahiert.
- Bei Prädiktionen, die über diesen Zeitraum hinausgehen, wird das Verfahren von McCarthy und Luzum (1990) angewendet, wobei allerdings auf eine Glättung der Daten verzichtet wird. Hier werden im Vorfeld ebenfalls die bekannten Einflüsse subtrahiert. Der Unterschied zu obiger Reduktion besteht darin, dass nun anstatt des linearen ein quadratischer Verlauf modelliert wird.

Für die Parameterschätzung der bekannten Einflüsse werden in beiden Fällen die Daten der jeweils letzten 1500 Tage herangezogen. Die gemittelten Ergebnisse sind in nachstehender Aufstellung zusammengefasst:

Vorhersagetag	10	20	30	40	60	90	120	150	180
RMS^{UT1} [ms]	0.9	2.5	4.3	6.0	8.7	14.4	16.5	22.5	29.6

6.1.5 Vorhersagegenauigkeiten nach Malkin (1999)

In der Veröffentlichung von Malkin (1999) werden zwei bereits existierende Vorhersagemethoden miteinander verglichen:

Internationaler Dienst	Datenreihe	Vorhersagemethode
IAA EOP Service	(EOP)IERS C04	Malkin und Skurikhina (1996)
IERS Subbureau for Rapid Service and Prediction (USNO)	NEOS	McCarthy und Luzum (1990)

Zur Bedeutung der noch nicht erwähnten Abkürzungen:

USNO: U.S. Naval Observatory

NEOS: National Earth Orientation Service (der U.S.A)

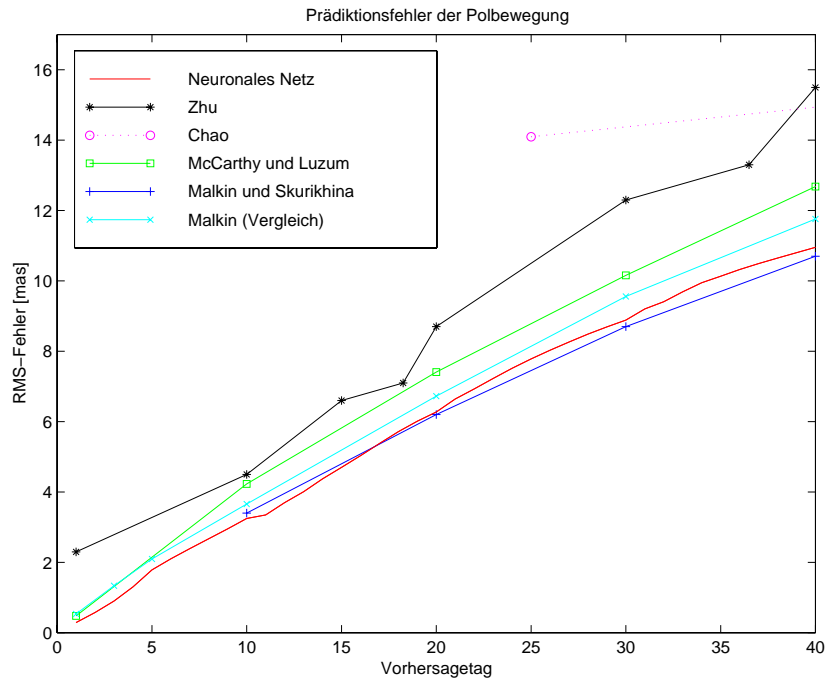
Die Berechnung der Genauigkeiten erfolgt bei diesem Vergleich nicht den jeweiligen Vorhersagemethoden der Autoren folgend, sondern in leicht veränderter Form: Die Ungenauigkeiten der letzten vorhandenen, aktuellen Werte fließen zusätzlich in die Berechnung der RMS-Fehler mit ein. Dadurch wird der Tatsache Rechnung getragen, dass aus den jüngsten Beobachtungen nicht endgültige Werte errechnet und veröffentlicht werden, sondern dass diese Werte im Laufe der Zeit noch angepasst werden. Diese Genauigkeitsberechnung hat gegenüber der herkömmlichen Methode den Vorteil, dass sie näher an der wirklichen Prädiktionsgenauigkeit liegt.

Nachfolgende Tabelle bietet eine Auswahl der erzielten Ergebnisse:

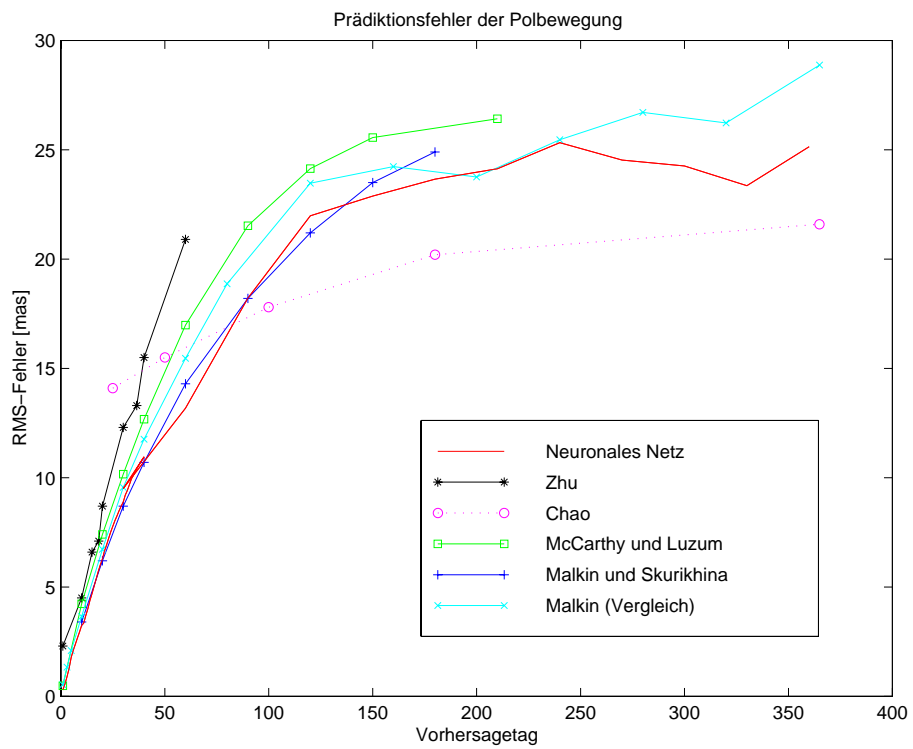
Vorhersagetag		1	5	10	20	40	80	180	280	365
RMS^{pol} [mas]	IAA	0.59	2.21	3.82	6.94	11.66	18.13	21.68	27.48	22.46
	USNO	0.48	1.98	3.50	6.51	11.86	19.61	25.07	25.94	35.29
RMS^{UT1} [ms]	IAA	0.18	0.59	1.44	3.37	7.06	15.92	42.87	100.46	166.81
	USNO	0.10	0.50	1.38	3.19	6.90	16.66	48.90	111.29	188.34

6.2 Bewertung und Einordnung des Verfahrens mit neuronalen Netzen

Um die Vorhersagequalität, die wir mit neuronalen Netzen erreichen, einordnen und bewerten zu können, vergleichen wir die mit der Datenreihe C04 erreichte Ergebnisse mit den in Kapitel 6.1 vorgestellten alternativen Vorhersagemethoden. Die Abbildungen 6.1 und 6.2 zeigen den graphischen Verlauf der Fehlerkurven im Vergleich.

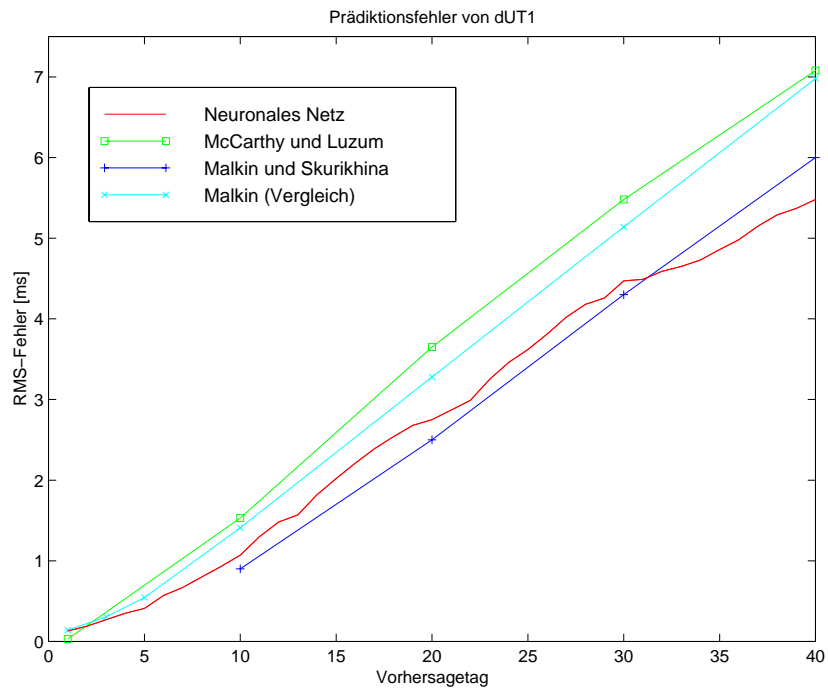


(a) Vorhersagetage 1 bis 40

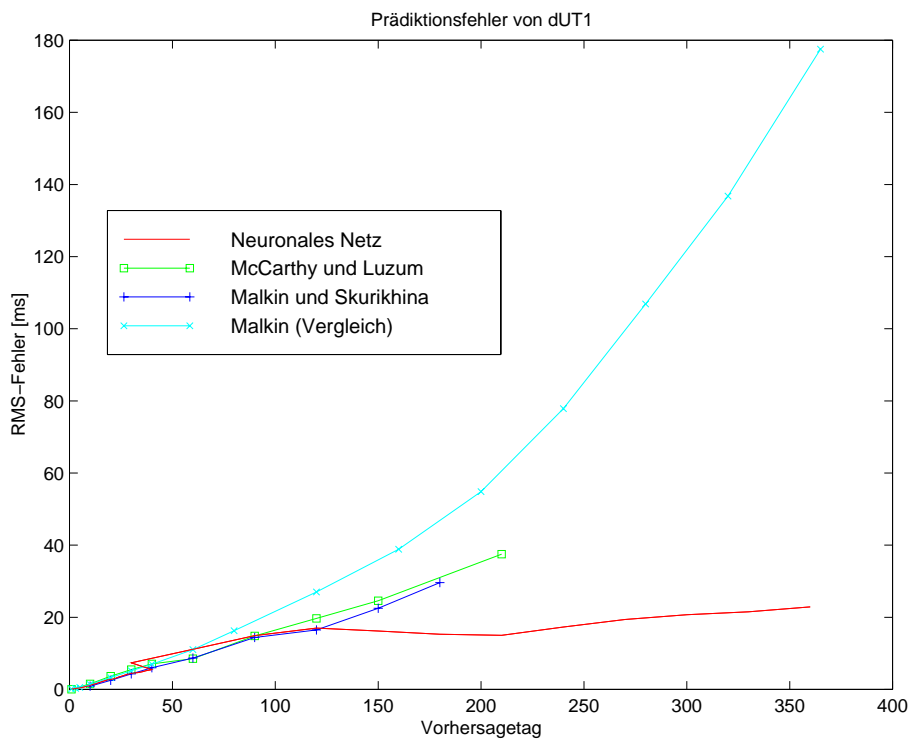


(b) Vorhersagetage 1 bis 365

Abbildung 6.1: Vergleich verschiedener Methoden zur Vorhersage der Polbewegung mit dem Resultat der neuronalen Netze



(a) Vorhersagetage 1 bis 40



(b) Vorhersagetage 1 bis 365

Abbildung 6.2: Vergleich verschiedener Methoden zur Vorhersage von $\Delta UT1$ mit dem Resultat der neuronalen Netze (die Unstetigkeit zwischen dem 30. und 40. Vorhersagetag ist auf die separate Vorhersage des kurz- und mittelfristigen Verlaufes zurückzuführen)

Die Fehlerkurve von Chao (1984) ist gestrichelt dargestellt, da sie nicht direkt mit den übrigen Methoden verglichen werden darf (siehe Kapitel 6.1.2).

Auf den ersten Blick können wir feststellen, dass die Vorhersage mit neuronalen Netzen nicht nur mit den gebräuchlichen Methoden mithalten kann, sondern teilweise sogar die besten Ergebnisse liefert.

6.2.1 Polbewegung

Die in dieser Arbeit erlangte Prädiktionsgenauigkeit für die kurzfristige Vorhersage der Polbewegung innerhalb von 40 Tagen ist vergleichbar mit den Ergebnissen von Malkin und Skurikhina (1996), deren Verfahren in der Gegenüberstellung am besten abschneidet. Das Resultat der mittelfristigen Vorhersage bis zu einem Jahr bringt sogar für die meisten Vorhersagetage die besten Ergebnisse.

Vergleichbar mit unserer Vorgehensweise zur Datenreduktion werden in den meisten Vorhersagemethoden alle bekannten Komponenten der Polbewegung modelliert. Dieses Modell wird anschließend extrapoliert. Während Zhu (1981, 1982) von einer konstanten Chandlerschen Periode ausgeht und damit die schlechtesten Ergebnisse erzielt, sind die Resultate von McCarthy und Luzum (1990) durch die Berücksichtigung einer variablen Periodendauer deutlich besser. Malkin und Skurikhina (1996) steigern die Genauigkeit dieses Verfahrens weiter: Durch ARIMA werden die Fehler im Vorhersagezeitraum bis zu 90 Tagen merklich reduziert. Die Anpassung der von Malkin und Skurikhina (1996) verwendeten linearen Gewichtsfunktion bringt bei Vorhersagen, die 90 Tage überschreiten, deutliche Verbesserungen.

Die von uns angewendete Reduktion der Polbewegung um die bekannten Einflüsse ist am ehesten mit der Modellierung nach McCarthy und Luzum (1990) vergleichbar, da auch wir die Periode der Chandlerschen Schwingung mitschätzen. Der Genauigkeitsunterschied zwischen dem Ergebnis von McCarthy und Luzum (1990) und unserem Verfahren ist also eindeutig den neuronalen Netzen zuzuschreiben.

6.2.2 Erdrotation

Ähnlich verhält es sich mit $\Delta UT1$. Auch hier sind die Verbesserungen von Malkin und Skurikhina (1996) gegenüber der Methode nach McCarthy und Luzum (1990) deutlich zu erkennen. Die Vorhersage der kurzfristigen Schwankungen gelingt mit ARIMA sogar besser als mit der Methode der neuronalen Netze. Allerdings könnte die Genauigkeit in diesem Bereich durch die Umrechnung von ΔLOD in $\Delta UT1$ noch gesteigert werden.

Sehr auffällig ist vor allem der mit Abstand kleinste Fehler der mittelfristigen Vorhersage. Das in dieser Arbeit entwickelte Verfahren zeigt hier seinen größten Vorteil gegenüber den herkömmlichen Methoden.

Kapitel 7

Zusammenfassung und Ausblick

Konfigurationen und Ergebnisse

Zur Abrundung der Arbeit werden abschließend die gewonnenen Erkenntnisse über die optimalen Konfigurationen und die damit erzielten Ergebnisse noch einmal kurz zusammengefasst:

Vorhersagen bis zu einem Jahr profitieren beim Lernvorgang von einer a priori Reduktion der Zeitreihen um die bekannten Einflüsse. Die Polbewegung wird deshalb um die jährliche Schwingung, die Chandlersche Schwingung und einen linearen Anteil reduziert. Bei $\Delta UT1$ und ΔLOD wird neben dem linearen Anteil und der jährlichen Schwingung die halbjährliche Komponente berücksichtigt. Zusätzlich wird der Einfluss der Gezeiten modelliert. Selbstverständlich müssen bei $\Delta UT1$ vorab die Schaltsekunden subtrahiert werden, um eine stetige Funktion zu erhalten.

Zur Vorhersage der Erdrotationsparameter erweisen sich feedforward-Netze mit shortcut-Verknüpfungen in Verbindung mit Vorhersagestrategie 3 (vgl. Kapitel 3.2.1) als am besten geeignet. Als Aktivierungsfunktion verwenden wir die logistische Funktion, als Ausgabefunktion die Identität. Vor dem Trainingsstart werden alle Gewichte mit Zufallszahlen aus dem Intervall $[-1;1]$ initialisiert. Die Rückführung des Netzausgabebefehlers und die damit verbundene Gewichts Anpassung während des Trainings geschehen in topologischer Reihenfolge, d.h. von der Ausgabeschicht hin zur Eingabeschicht. Eine zufällige Auswahl der Muster aus dem Trainingsdatensatz führt beim Lernen zu einer erheblichen Beschleunigung des Trainingsfortschritts im Vergleich zu einer zeitlich geordneten Musterpräsentation. Ein Testdatensatz am Anfang der Datenreihe, der die ersten 10 Prozent aller Muster beinhaltet, dient zur Kontrolle der Extrapolationsfähigkeit und liefert damit die Abbruchbedingung für das Training.

Die Vorhersage der ersten 40 Tage geschieht in vier Etappen. Das konstante Eingabemuster $[1 \dots 10 \ 12 \ 16 \ 24 \ 40 \ 72 \ 136]$ steht dabei den vier Ausgabemustern $[1 \dots 10]$, $[11 \dots 20]$, $[21 \dots 30]$ und $[31 \dots 40]$ gegenüber. Ein neuronales Netz mit der Topologie 16-8-4-10 dient jeweils als Basis. Eine Besonderheit bei der Polbewegung liegt in der gemeinsamen Vorhersage der x- und y- Komponente für die ersten fünf Tage. Hierfür verwenden wir die Netztopologie 32-32-16-20. Auch durch die Verwendung der summierten ΔLOD -Werte zur Vorhersage der ersten 8 Tage von $\Delta UT1$ kann ein Genauigkeitsgewinn erreicht werden. Dieser ist allerdings in den Ergebnissen nicht berücksichtigt, da bei der Umrechnung von ΔLOD nach $\Delta UT1$ Diskretisierungsfehler auftreten, die durch die begrenzte Auflösung der Datenreihen verursacht werden.

Auch bei der Vorhersage der mittelfristigen Entwicklung bringt eine Datenreduktion große Vorteile. Da sich die Prädiktion jetzt allerdings bis zu einem Jahr in die Zukunft erstreckt, dehnen wir das Eingabemuster sinnvollerweise weiter in die Vergangenheit aus als bei der Vorhersage der ersten 40 Tage. Das Eingabemuster $[1 \ 30 \ 60 \ 90 \ 120 \ 150 \ 180 \ 210 \ 240 \ 270 \ 300 \ 330 \ 360 \ 420 \ 540 \ 780 \ 1260]$ überdeckt somit einen Zeitbereich von ca. $3\frac{1}{2}$ Jahren. Im Ausgabemuster $[30 \ 60 \ 90 \ 120 \ 150 \ 180 \ 210 \ 240 \ 270 \ 300 \ 330 \ 360]$ sind die zukünftigen Werte im Monatsabstand gespeichert. Auch zur Prädiktion der mittelfristigen Entwicklung eignen sich zwei versteckte Schichten mit 8 und 4 Neuronen.

Wie in Kapitel 6.2 gezeigt, können mit den oben beschriebenen Konfigurationen Ergebnisse erzielt werden, die bisherigen Vorhersagemethoden gleichwertig sind, diese sogar zum größten Teil übertreffen. Die Genauigkeit des in dieser Arbeit entwickelten Verfahrens soll durch die Übersicht der RMS-Fehler nochmals zum Ausdruck gebracht werden:

Vorhersagetag	RMS^{pol} [mas]	$RMS^{\Delta UT1}$ [ms]	$RMS^{\Delta LOD}$ [ms]
1	0.29	0.13	0.019
2	0.57	0.19	0.049
3	0.95	0.27	0.074
4	1.30	0.35	0.097
5	1.79	0.41	0.121
6	2.10	0.57	0.142
7	2.39	0.67	0.159
8	2.67	0.80	0.174
9	2.95	0.93	0.184
10	3.25	1.07	0.193
15	4.70	2.02	0.246
20	6.28	2.75	0.251
25	7.78	3.62	0.249
30	8.89	4.47	0.245
35	10.14	4.86	0.263
40	10.96	5.48	0.258
60	13.16	11.14	0.292
90	18.21	14.91	0.306
120	21.93	16.98	0.314
150	22.92	16.21	0.330
180	23.67	15.27	0.361
210	24.14	15.1	0.397
240	25.30	17.26	0.377
270	24.51	19.42	0.386
300	24.27	20.69	0.402
330	23.36	21.50	0.372
360	25.09	22.94	0.347

Beim Einsatz in der Praxis steht von Tag zu Tag mehr Trainingsinformation zur Verfügung. Dadurch kann mit einer kontinuierlichen Genauigkeitssteigerung gerechnet werden.

Problematik

Trotz der guten Ergebnisse darf die Problematik des rechtzeitigen Trainingsabbruches nicht übersehen werden. Das Minimum der Testdaten tritt im Allgemeinen nicht nach der gleichen Anzahl an Trainingsiterationen auf wie das Minimum der Validierungsdaten. Somit entspricht das Vorhersageresultat fast nie der bestmöglichen Prädiktion. Der Vergleich der Vorhersagen für $\Delta UT1$ einerseits mit der C04- und andererseits mit der SPACE-Reihe verdeutlicht, dass in der Praxis durchaus mit mehr oder weniger großen Genauigkeitsschwankungen gerechnet werden muss. Ähnlich einem chaotischen Verhalten reagiert das neuronale Netz auf kleine Unterschiede in den Trainingsdaten offensichtlich sehr sensibel.

Ausblick

Das Potential des Einsatzes neuronaler Netze zur Vorhersage der Erdrotation ist in dieser Arbeit deutlich geworden. Trotzdem konnte nur ein Teil der möglichen Varianten untersucht werden. Sicherlich sind noch weitere Verbesserungen vorstellbar. Daher werden abschließend noch einige Anregungen und Ideen beschrieben:

- Bei der Vorhersage der mittel- und langfristigen Entwicklung könnte eine variable Glättung Vorteile bringen. Je größer der Vorhersagezeitraum gewählt wird, desto weniger haben hohe Frequenzen Einfluss und desto stärker sollten diese eliminiert werden.
- Denkbar ist auch das Einbringen von Zusatzinformation in das Netz neben den Zeitreihen der Erdrotationsparameter selbst:

- Für das *Atmospheric Angular Momentum* (AAM) liegen aktuellere Werte vor, als für die Erdrotationsparameter selbst. Daneben sind auch Vorhersagen der zukünftigen 10 Tage ständig verfügbar. Da der zeitliche Verlauf des AAM sehr stark mit der Rotationsgeschwindigkeit der Erde und mit der Polbewegung korreliert ist, kann durch die Berücksichtigung dieser Daten die Qualität der kurzfristigen Vorhersagen der Erdrotationsparameter erhöht werden (Freedman et al., 1994). Zusätzliche Eingabe-neuronen könnten diese Information in das neuronale Netz mit einbringen und somit die Vorhersage kurzfristiger Schwankungen erleichtern.
- Aus den ursprünglichen Zeitreihen können weitere Daten abgeleitet werden. So ist es durchaus vorstellbar, dass der Trainingsverlauf durch Einbringen der ersten oder zweiten Ableitung oder verschiedener statistischer Größen positiv beeinflusst wird.
- Es könnte der Versuch unternommen werden, alle Erdrotationsparameter als Eingabeinformation an nur ein gemeinsames Netz anzulegen. Auf der einen Seite können dadurch Korrelationen zwischen den Parametern eine Vorhersage begünstigen. Auf der anderen Seite dürfte es sich schwierig gestalten, eine gemeinsame Lernfunktion zu finden, da das Trainingsverhalten der Erdrotationsparameter bei separaten Netzen sehr unterschiedlich ausfällt.
- Die Verwendung von GPS-Reihen an Stelle kombinierter Lösungen wie C04 oder SPACE dürfte aufgrund ihrer hohen Aktualität ebenfalls interessante Ergebnisse liefern. Besonders bei der kurzfristigen Vorhersage sind hier Genauigkeitssteigerungen zu erwarten.
- Speziell für die langfristige Vorhersage der Polbewegung ist noch ein alternativer Weg gangbar: Nicht die Werte der x- und y-Komponenten selbst sind dabei Gegenstand der Vorhersage, sondern lediglich abgeleitete Parameter. Diese könnten z.B. die geschätzten Unbekannten unserer Variationsanalyse aus Kapitel 4 darstellen: Nach der Vorhersage sowohl der Amplituden, Phasen und Frequenzen der jährlichen Schwingung und der Chandlerschen Bewegung als auch der Parameter des linearen Trends können die Funktionswerte der x- und y-Komponente anschließend wieder rekonstruiert werden.
- Das in dieser Arbeit entwickelte Verfahren basiert auf der Aneinanderreihung mehrerer Teilvorhersagen. Die dabei auftretenden Unstetigkeiten zwischen den einzelnen Vorhersageabschnitten könnten durch Einsatz einer linearen Übergangsfunktion bereinigt werden.
- Um die Genauigkeitssteigerung der beobachteten Erdrotationsparameter innerhalb der letzten Jahre zu berücksichtigen, könnten die Muster des Trainingsdatensatzes unterschiedlich oft trainiert werden. Genauere Daten müssten höher gewichtet und dem Netz während des Trainings demnach häufiger präsentiert werden als Daten mit höherer Standardabweichung. Da diese Möglichkeit von SNNS nicht unterstützt wird, könnten wir uns damit behelfen, genauere Daten proportional zur Genauigkeit im Musterdatensatz mehrfach zu speichern.

Kapitel 8

Anhang: Entwickelte Software

Auf der beiliegenden CD-ROM sind alle entwickelten Programme gespeichert, die neben dem SNNS-Softwarepaket zur Durchführung der in dieser Arbeit anfallenden Aufgaben nötig sind. Tabelle 8.2 bietet einen Überblick über die wichtigsten ausführbaren Programme. Die Sortierung der Aufstellung entspricht der logischen Reihenfolge, in der die Programme angewendet werden. In der zweiten Spalte mit der Bezeichnung “Typ” ist die Programmart verschlüsselt:

C_W: unter DOS und WINDOWS ausführbares Programm in C oder C++

C_L: unter Linux ausführbares Programm in C oder C++

M: unter MATLAB ausführbares M-File

b: ausführbares Batch-Programm unter LINUX

B: BATCHMAN-Programm

Auf eine Auflistung und Beschreibung weiterer Programme z.B. zur graphischen Ausgabe von Funktionen, Fehlerkurven oder Differenzen von Datenreihen, zu Konvertierungszwecken oder zur Erleichterung und Validierung der Programmierarbeit wird verzichtet. Ebenso fehlen in der Übersicht Funktionen, die separat als Datei gespeichert wurden und von den ausführbaren Programmen aufgerufen werden. Diese enthalten u.a. verschiedene Ausgleichungsmechanismen, Schaltsekunden- und Gezeitenmodellierung, Fouriertransformation, Berechnungen zu Kugelflächenfunktionen, Zeittransformationen oder Kernroutinen zur Schnittstellenverwaltung zwischen MATLAB und SNNS.

Dateiname	Typ	Beschreibung
c04ext.cpp	C_W	Konvertierung der C04-Daten vom IERS in ein von MATLAB lesbares Format
jpl2c04.m	M	Konvertierung der SPACE-Reihe des JPL in ein von MATLAB lesbares Format (analog zu C04)
get_seas_pol.m get_seas_lod.m get_seas_ut1.m	M	Variationsanalyse der Polbewegung, von Δ LOD und Δ UT1
red_pol.m red_lod.m red_ut1.m	M	FFT, Reduktion der Datenreihen um alle bekannten Einflüsse und graphische Darstellung
gen_pat.m	M	Datennormierung, Mustergenerierung, Musteraufteilung, Konvertierung der reduzierten Datenreihen ins PAT-Format und Erstellung einer LOG-Datei für die spätere Analyse
gen_pat2.m	M	Entspricht "gen_pat.m", für gemeinsame Vorhersage zweier Datenreihen (z.B. x und y) mit nur einem neuronalen Netz
start_bat	b	Startet den Trainingsvorgang mit allen Trainingsprogrammen
train.bat	B	Trainingsprogramm: Parametereinstellung, numerische Überwachung, Ausgabe des trainierten Netzes, der RES-Dateien und des protokollierten Trainingsverlaufes (pro Konfiguration existiert ein Trainingsprogramm)
start_plot	b	Überwachung des Trainingsfortschrittes während des Lernens durch graphische Darstellung der Netzausgabefehler
res2pat2.cpp	C_L	Einlesen der Programmparameter aus "parameters.par" und Start der sukzessiven Vorhersage durch automatische Generierung von PAT-Dateien aus vorangegangenen RES-Dateien
iterat.bat	B	Sukzessive Vorhersage (wird von "res2pat2.cpp" aufgerufen)
res2mat_start.m	M	Einlesen der RES-Dateien, der LOG-Datei und des protokollierten Trainingsverlaufes, Auswertung des Ergebnisses mit Hilfe einer Vielzahl von numerischen und graphischen Analysemöglichkeiten
rei2mat_start.m	M	Entspricht "res2mat_start", für die sukzessive Vorhersage
res2mat2_start.m	M	Entspricht "res2mat_start", für die Auswertung der gleichzeitigen Vorhersage zweier Datenreihen mit nur einem neuronalen Netz (z.B. x und y)
res2mat_pred.m	M	Auswertung für den Anwendungsfall: kurz- und mittelfristige Vorhersage
res2mat2_pred.m	M	Auswertung für den Anwendungsfall: langfristige Vorhersage

Tabelle 8.2: Auswahl der erstellten Software

Literaturverzeichnis

- [1] Ballani L., Barthelmes F., Greiner-Mai, H., Jochmann, H., Klees R. und Lehmann R., Erdrotationschwankungen, Erdkerndynamik und Schwerefeld, Zweijahresbericht 1992/1993, GeoForschungsZentrum Potsdam, 1993.
- [2] Brosche, P. und Schuh, H., Tides and Earth rotation, *Surveys in Geophysics* 19, 417–430, 1998.
- [3] Chao, B.F., Predictability of the Earth's polar motion, *Bull. Géod.* 59, 81–93, 1984.
- [4] Conway, A.J., Time Series, Neural Networks and the Future of the Sun, Department of Physics and Astronomy, University of Glasgow, 1998.
- [5] Cybenko, G., Mathematics of Control, Signal and Systems, 1989.
- [6] Drewes, H., Demonstrationsmaterial zur Vorlesung Geodätische Geodynamik, Technische Universität München, Deutsches Geodätisches Forschungsinstitut, 1998.
- [7] Ebner, H., Skript zur Vorlesung Ausgleichsrechnung I und II, Lehrstuhl für Photogrammetrie und Fernerkundung, Technische Universität München, 1998.
- [8] Egger, D., Neuronales Netz prädiziert Erdrotationsparameter, *Allgemeine Vermessungsnachrichten (AVN)* 11/12, 517–524, 1992.
- [9] Egger, D., Fröhlich, H., Prädiktion von Erdrotationsdaten - klassisch und neuronal, Forschungseinrichtung Satellitengeodäsie, Institut für Astronomische und Physikalische Geodäsie, Technische Universität München, 1992.
- [10] Freedman, A.P., Steppe, J.A., Dickey, J.O., Eubanks, T.M. und Sung, L.-Y., The short-term prediction of universal time and length of day using atmospheric angular momentum, *Journal of Geophysical Research*, Vol. 99, No. B4, 6981–6996, April 1994.
- [11] Gipson, J.M., Very long baseline interferometry determination of neglected tidal terms in high-frequency Earth orientation variation, *Journal of Geophysical Research*, Vol. 101, NO. B12, 28 051–28 064, Dezember 1996.
- [12] Gross, R.S., Combinations of Earth Orientation Measurements: SPACE98, COMB98 and POLE98, Jet Propulsion Laboratory (JPL), JPL Publication 99-6, California, April 1999.
- [13] Haas, R., Transformation of notations of models of diurnal and semidiurnal variations in Earth's Rotation, GIUB VLBI Technical Memorandum Nr. 1, 1995.
- [14] Heine, K., Beschreibung von Deformationsprozessen durch Volterra- und Fuzzy-Modelle sowie Neuronale Netze, Deutsche Geodätische Kommission, Dissertation, C, 516, München, 1999.
- [15] Hoffmann, N., Kleines Handbuch neuronale Netze, Braunschweig, Wiesbaden, Vieweg, 1993.
- [16] Husfeld, D., Astronomische Zeitmessung, <http://step.iapg.verm.tu-muenchen.de/users/bene/index.html>, 1996.
- [17] International Earth Rotation Service (IERS), 1998 IERS Annual Report, Central Bureau of IERS, Observatoire de Paris, 1999.
- [18] Kinnebrock, W., Neuronale Netze, Grundlagen, Anwendungen, Beispiele, R. Oldenbourg Verlag München Wien, 1994.
- [19] Malkin, Z., On Estimation of Real Accuracy of EOP Prediction, IAU Colloquium 187, Cagliari, September 1999.

- [20] Malkin, Z. und Skurikhina, E., On Prediction of EOP, Comm. IAA, No 93, 1996.
- [21] McCarthy, D.D. und Luzum, B.J., Prediction of Earth orientation, Bulletin Godesique 65, 18–22, 1991.
- [22] McCarthy, D.D. (ed.), IERS Technical Note 21, IERS Conventions (1996), Observatoire de Paris, 1996.
- [23] Moll, H., Entwicklung und Test eines Computerprogrammes zur Vorhersage von Zeitreihen mittels neuronaler Netze, Diplomarbeit, Institut fur Astronomische und Physikalische Geodasie, Technische Universitat Munchen, 1999.
- [24] Muller, J., Zeitskalen, Mitteilungen des Bundesamtes fur Kartographie und Geodasie, Band 5, Frankfurt, 77–83, 1999.
- [25] Otto, P., Identifikation nichtlinearer Systeme mit Kunstlichen Neuronalen Netzen, at - Automatisierungstechnik 43 (1995) 2, R. Oldenbourg Verlag Munchen, 62–68, 1995.
- [26] Ray, R.D., Steinberg, D.J., Chao, B.F. und Cartwright, D.E., Diurnal and Subdiurnal Variations in the Earth's Rotation Rate Induced by Oceanic Tides, Science, 264, 830–832, 1994.
- [27] Ritter, H., Martinetz, T. und Schulten, K., Neuronale Netze, Eine Einfuhrung in die Neuroinformatik selbstorganisierender Netzwerke, Addison-Wesley, 1990.
- [28] Rummel, R., Physical Geodesy I, Fakultat der Geodasie, Technische Universitat Delft, Oktober 1992.
- [29] Schuh, H., Zur Spektralanalyse von Erdrotationsschwankungen, Die Arbeiten des Sonderforschungsbereiches 78 Satellitengeodasie der Technischen Universitat Munchen, 1980; Veroffentlichungen der Bayerischen Kommission fur die Internationale Erdmessung der Bayerischen Akademie der Wissenschaften, Heft Nr. 41, 176–193, 1981.
- [30] Seiler, U., Wunsch, J., A refined model for the influence of ocean tides on UT1 and polar motion, Astron. Nachr. 316, 6, 419–423, 1995.
- [31] Stocker, H., Taschenbuch der Physik: Formeln, Tabellen, Ubersichten, Frankfurt am Main, Verlag Harri Deutsch, 1994.
- [32] Yoder, C.F., Williams, J.G., und Parke, M.E., Tidal variations of Earth rotation, J. Geophys. Res. 86, B2, 881–891, 1995.
- [33] Zebhauser, B., Realisierung von Bezugssystemen, Mitteilungen des Bundesamtes fur Kartographie und Geodasie, B5, Frankfurt, 70–76, 1999.
- [34] Zell, A., Simulation Neuronaler Netze, Addison-Wesley, 1994.
- [35] Zell, A, Mamier G., Vogt, M., Mache, N., Hubner, R., Doring, S., Hermann, K.U., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., Wieland, J., SNNS, Stuttgart Neural Network Simulator, User Manual, Version 4.1, 1995.
- [36] Zhu, S.Y., Prediction of Earth rotation and polar motion, Reports of the Department of Geodetic Science and Surveying, Report No. 320, 1981.
- [37] Zhu, S.Y., Prediction of polar motion, Bull. God. 56, 258–273, 1982.