

Optimierung und Analyse von Fachwerkstrukturen durch Neuronale Netze

Jürgen Streng

Optimierung und Analyse von Fachwerkstrukturen durch Neuronale Netze

Von der Fakultät Luft- und Raumfahrttechnik der
Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von Jürgen Streng aus Tuttlingen

Hauptberichter: Professor Dr.-Ing. habil. Bernd Kröplin

Mitberichter: Professor Dr.-Ing. habil. Dieter Dinkler

Tag der mündlichen Prüfung: 22. Dezember 2000

Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen
Universität Stuttgart

**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**

Forschungsbericht 2001-06

Optimierung und Analyse
von Fachwerkstrukturen
durch Neuronale Netze

Jürgen Streng

Institut für Bauweisen-
und Konstruktionsforschung
Stuttgart

180 Seiten

34 Bilder

32 Tabellen

132 Literaturstellen



Manuskript eingereicht im August 2001.

Erscheint gleichzeitig als Dissertation an der Fakultät Luft und Raumfahrttechnik
der Universität Stuttgart.

Optimierung und Analyse von Fachwerkstrukturen durch Neuronale Netze

Deutsches Zentrum für Luft- und Raumfahrt

Institutscluster Werkstoffe und Strukturen
Institut für Bauweisen- und Konstruktionsforschung
Pfaffenwaldring 38-40
70569 Stuttgart

Stuttgart, im Dezember 2000

Institutsleiter:

Prof. Dipl.-Ing. Richard Kochendörfer

Verfasser:

Dipl.-Ing. Jürgen Streng

Abteilungsleiter:

Dr.-Ing. Kuno K.U. Stellbrink

Problemspezifische Neuronale Netze, Fachwerkstrukturen, mehrgliedrige Evolutionsstrategie, Mehrzieloptimierung, *Pruning* Neuronaler Netze, Schadensanalyse, Lösung inhomogener linearer Gleichungssysteme

Optimierung und Analyse von Fachwerkstrukturen durch Neuronale Netze

Übersicht

Die in dieser Arbeit neu entwickelte Systematik einer Multikriterienoptimierung ebener Fachwerkstrukturen basiert auf der Integration Neuronaler Netze in den Algorithmus einer mehrgliedrigen Evolutionsstrategie. Die trainierten konventionellen und problemspezifischen Neuronalen Netze erfassen Kriterien und Restriktionen der zu untersuchenden Optimierungsaufgabe. Optimierungssimulationen bestätigen die Funktionsfähigkeit dieser neuen Systematik. Am Beispiel statisch bestimmter und unbestimmter Fachwerkstrukturen werden Möglichkeiten einer selbstadaptiven Schadensanalyse durch trainierte sowie durch *Pruning* reduzierte Neuronale Netze untersucht. Dies ermöglicht die Bestimmung eines unbekanntem inversen Zusammenhangs. Weiterhin wird ein auf problemspezifischen Neuronalen Netzen beruhendes, neuartiges Lösungsverfahren inhomogener linearer Gleichungssysteme vorgestellt.

Problemspecific neural networks, multimembered evolutionary strategy, multicriteria optimization, pruning of neural networks, damage analysis, solution of inhomogeneous linear equation systems using problemspecific neural networks

Optimization and analysis of truss structures using neural networks

Abstract

The integration of neural networks and a multimembered evolutionary strategy leads to a new multicriteria optimization approach for plane truss structures. Conventional as well as problemspecific neural networks describe the appertaining restrictions and criteria of the investigated optimization problem. Advantages of this new approach are the minimal response time required by trained neural networks in the context of a stochastic optimization method as well as a probability greater than zero to find global extrema in multimodal optimization problems.

Conventional neural networks are able to assess the impairment of damaged structural

systems. The fully connected starting topologies of these networks are minimized with a combination of pruning algorithms and conventional network training. Statically determinate and indeterminate systems are investigated with a main focus on indeterminate truss structures and related problems concerning sufficient data diversification and adequate data preprocessing. As a result neural networks are able to solve specific inverse functional relations, and thereby assess the impairment of a damaged structural system with a minimized network topology.

This research work also presents a new methodology for the solution of inhomogeneous linear equation systems using simple problemspecific neural networks.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit am Institut für Bauweisen und Konstruktionsforschung des Deutschen Zentrums für Luft- und Raumfahrt e.V. in Stuttgart.

Herrn Professor Dr.-Ing. D. Dinkler vom Institut für Statik der Technischen Universität Braunschweig sowie Herrn Professor Dr.-Ing. B. Kröplin von Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen der Universität Stuttgart danke ich für die Bereitschaft, meine Arbeit zu betreuen. Ihre Diskussionsbereitschaft und kritische Unterstützung hat sehr zum Gelingen der Arbeit beigetragen. Für die Übernahme des Prüfungsvorsitzes danke ich Herrn Professor Dipl.-Ing. R. Voit-Nitschmann vom Institut für Flugzeugbau der Universität Stuttgart.

Mein besonderer Dank gilt Herrn Dr.-Ing. Kuno K.U. Stellbrink und meinen Kollegen beim DLR Stuttgart für ihre Unterstützung dieses Dissertationsvorhabens.

Ein Teil der vorliegenden Arbeit entstand während meines Aufenthalts als Gastwissenschaftler des Department of Aeronautics and Astronautics der Stanford University, USA. Der Auslandsaufenthalt wurde ermöglicht durch ein Doktorandenstipendium des Deutschen Akademischen Austauschdienstes aus Mitteln des zweiten Hochschulsonderprogramms des Bundesministeriums für Forschung und Technologie der Bundesrepublik Deutschland sowie durch die Unterstützung des Department of Aeronautics and Astronautics der Stanford University. Ich möchte mich bei Professor I. Kroo und den Mitgliedern der Aircraft Aerodynamics and Design Group für die freundliche Aufnahme, die interessanten Diskussionen und die mir entgegengebrachte Hilfsbereitschaft bedanken.

Mein Dank gilt weiter der Flughafen Frankfurt Main Stiftung. Sie förderte nach Auslaufen meines DLR-Vertrags die Fertigstellung der Dissertation durch ein Stipendium.

Herrn Dr.-Ing. S. Gollwitzer von der Firma RCP GmbH in München möchte ich für seine Hilfsbereitschaft und die kostenlose Bereitstellung des kommerziellen Software-Pakets COMREL-TI Version 7(β) zur Validierung meiner Zuverlässigkeitsergebnisse danken. Herrn Dr.-Ing. H.P. Mlejnek vom Institut für Computeranwendungen der Universität Stuttgart danke ich für die wertvollen Hinweise nach der Durchsicht der Arbeit.

Stuttgart-Vaihingen, im Dezember 2000.

Ich widme diese Arbeit meiner Mutter und meinem Vater

Inhaltsverzeichnis

Vorwort	III
Verzeichnis der verwendeten Symbole	XIII
1 Einleitung	1
1.1 Aufgabenstellung	1
1.2 Stand der Forschung	3
1.2.1 Neuronale Netze in der Strukturmechanik	3
1.2.2 Neuronale Netze zur Zielfunktionsbeschreibung in der Optimierung .	4
1.2.3 Schadensanalyse in Fachwerkstrukturen	5
1.2.4 Korrelation Struktur – Neuronale Netze	7
2 Mathematik	9
2.1 Grundlagen der Optimierung	9
2.1.1 Definitionen und Vereinbarungen	9
2.1.2 Restriktionen	11
2.2 Optimierungsverfahren	13
2.2.1 Überblick über deterministische Verfahren	13
2.2.2 Überblick über stochastische Verfahren	14
2.3 Dimensionsanalyse	15
3 Tragwerksoptimierung	18
3.1 Grundlagen	18
3.1.1 Strukturdefinitionen und Vereinbarungen	18
3.1.2 Optimierung	19
3.2 Ersatzzielfunktionen für Vektoroptimierungsprobleme	20
3.2.1 Vektoroptimierungsproblem	20
3.2.2 Ersatzzielfunktion	21

3.3	Mehrgliedrige Evolutionsstrategie	23
3.3.1	Algorithmus der mehrgliedrigen Evolutionsstrategie	24
3.3.2	Programm zur mehrgliedrigen Evolutionsstrategie	25
3.4	Gegenüberstellung deterministischer und stochastischer Verfahren	27
4	Zuverlässigkeitstheorie	31
4.1	Grundlagen der Wahrscheinlichkeitsrechnung	31
4.2	Aufgaben der Statistik	33
4.3	Zuverlässigkeitstheorie	33
4.4	FORM	36
4.4.1	Methodik	36
4.4.2	Rosenblatt-Transformation	37
4.4.3	Approximation der Versagenswahrscheinlichkeit	38
4.5	Kopplung von mechanischem und stochastischem Modell	39
5	Neuronale Netze	42
5.1	Grundlagen	42
5.2	Netztopologien	44
5.2.1	Nomenklatur	45
5.2.2	Konventionelle Neuronale Netze	48
5.2.3	Problemspezifische Neuronale Netze	49
5.2.4	Multivariabler Summen-Produkt-Ansatz	52
5.3	Test- und Trainingsdaten	53
5.3.1	Erzeugung von Test- und Trainingsdaten	54
5.3.2	Permutationsalgorithmus	54
5.3.3	Datenvorverarbeitung durch lineare Transformation	55
5.4	Netztraining	56
5.4.1	Training und Test	57

5.4.2	Backpropagation with Momentum Term and Flat Spot Elimination . .	58
5.4.3	Resilient Propagation	59
5.5	Pruning	60
5.5.1	Ausdünnungsverfahren	61
5.5.2	Optimal Brain Surgeon	63
5.5.3	Diskussion des OBS-Algorithmus	65
5.5.4	Realisation von OBS in SNNS	66
5.6	Leistungsfähigkeit von Feedforward-Netzwerken	67
5.6.1	Repräsentationsfähigkeit, Lernfähigkeit und Komplexität	67
5.6.2	Generalisierungsleistung	67
6	Neuronale Netze in der Strukturmechanik	69
6.1	Beschreibung von Systemantworten mit problemspezifischen Neuronalen Netzen	70
6.1.1	Dimensionsanalyse	70
6.1.2	Charakteristische Kenngrößen winkeltreuer ebener Fachwerkstrukturen	71
6.1.3	Datenvorverarbeitung	72
6.1.4	Test- und Trainingsdaten	74
6.2	Training problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Knotenverschiebung	75
6.2.1	Datenvorverarbeitung	76
6.2.2	Ansatzfunktionen	77
6.2.3	Training und Ergebnisse	79
6.3	Training problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Stabspannung	83
6.3.1	Datenvorverarbeitung	83
6.3.2	Ansatzfunktionen	84
6.3.3	Training und Ergebnisse	85

6.4	Pruning problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Knotenverschiebung	87
6.5	Pruning problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Stabspannung	91
6.6	Fazit	94
7	Neuronale Netze und Evolutionsstrategien zur Untersuchung von Mehrzieloptimierungsaufgaben	96
7.1	Grundprinzip	97
7.2	Beschreibung von Strukturzuverlässigkeiten durch Neuronale Netze	99
7.3	Einbindung von Neuronalen Netzen in Evolutionsstrategien	103
7.3.1	Berücksichtigung der Optimierungskriterien	104
7.3.2	Berücksichtigung der Optimierungsrestriktionen	104
7.3.3	Ersatzzielfunktion	105
7.4	Mehrzieloptimierung des Fachwerks FW2	105
7.4.1	Ersatzzielfunktion und Optimierungsrestriktionen	105
7.4.2	Optimierung des Fachwerks FW2 mit (10,100)-Evolutionsstrategie	107
7.5	Fazit	108
8	Schadensanalyse in Fachwerkstrukturen	110
8.1	Das inverse Problem	110
8.2	Schadensanalyse mit Neuronalen Netzen	112
8.2.1	Systematik	112
8.2.2	Limitierungen	114
8.2.3	Test- und Trainingsdaten	115
8.2.4	Netztopologien	117
8.3	Statisch bestimmte Fachwerke	117
8.3.1	Training der Ausgangsnetztopologien	120
8.3.2	Reduktion der Ausgangsnetztopologien	122

8.4	Statisch unbestimmte Fachwerke	125
8.4.1	Training und Reduktion der Ausgangsnetztopologien	127
8.5	Fazit	131
9	Analogie Struktur – Neuronale Netze	133
9.1	Neue Systematik	133
9.1.1	Mathematisches iLGS	136
9.1.2	Analogie Struktur – Problemspezifisches Neuronales Netz	138
9.2	Statisch bestimmte Fachwerke	140
9.3	Statisch unbestimmtes Fachwerk	142
9.4	Fazit	144
10	Zusammenfassung	147
	Literaturverzeichnis	149

Verzeichnis der wichtigsten verwendeten Symbole

a	[-]	Lösungsvektor eines iLGS
A	[-]	Dimensionsmatrix
$A_{j\ k}$	[mm ²]	Querschnittsfläche des Stabs $Stab_{j\ k}$
AF	[-]	Approximationsfehler
b	[-]	Vektor der Absolutglieder eines iLGS
d	[-]	Residuenvektor
\mathbb{D}	[-]	Definitionsbereich
$\det(\mathbf{X})$	[-]	Determinante der Matrix X
e	[-]	Anzahl der Trainingsepochen
E	[-]	1. Gesamtfehlerfunktion eines Neuronalen Netzes
E	[-]	2. Entwicklungspunkt
\mathbb{E}_U	[-]	Ereignisraum standardnormalverteilter Basisvariablen
$E_{j\ k}$	[$\frac{N}{\text{mm}^2}$]	Elastizitätsmodul des Stabs $Stab_{j\ k}$
Ez	[-]	Ersatzzielfunktion
f	[N]	Lastvektor
$f(\mathbf{x})$	[-]	Zusammenhang
f(x)	[-]	Zielfunktion einer Vektoroptimierungsaufgabe
F	[N]	Statischer Lastfall eines Fachwerks
f_j	[1]	Einzeloptimierungskriterium
f_{akt}	[-]	Aktivierungsfunktion eines Neurons
f_{aus}	[-]	Ausgabefunktion eines Neurons
f_{ein}	[-]	Eingabefunktion eines Neurons
f_{trans}	[-]	Transferfunktion eines Neurons
f_x	[-]	Wahrscheinlichkeitsdichtefunktion der Zufallsvariablen X
$f_{X,\log}$	[-]	Lognormalverteilung
F_x	[-]	Verteilungsfunktion der Zufallsvariablen X

$g(\mathbf{x})$	[-]	1. Gleichheits- bzw. Ungleichheitsrestriktionen
$g(\mathbf{Z}) = 0$	[-]	2. Grenzzustandsfunktion
$G_{j\ k}$	[N]	Reststeifigkeiten des Stabs $\text{Stab}_{j\ k}$
\mathbf{H}	[-]	Hesse-Matrix der Gesamtfehlerfunktion E
\mathbf{in}	[-]	Eingabevektor
$\inf f$	[-]	Infimum der Funktion f
\mathbf{K}	[-]	Exponentenmatrix
\mathbf{l}	[-]	Lernvektor
$l_{j\ k}$	[mm]	Länge des Stabs $\text{Stab}_{j\ k}$
L	[-]	Lagrange-Gleichung
L_q	[-]	<i>Saliency</i> (siehe Abschnitt 5.5.2)
$\text{limit}_{\delta_{k_n\ i,y}}$	[mm]	Zulässige Knotenverschiebung
$\text{limit}_{\sigma_{j\ k}}$	$[\frac{N}{\text{mm}^2}]$	Zulässige Stabspannung
\mathbb{M}	[-]	Zulässiger Bereich
$\{M_1, M_2, \dots, M_m\}$	[-]	Grundgrößensystem
M_{FW}	[kg]	Masse des Fachwerks FW
mmod	[-]	Mechanisches Modell
\mathbb{N}	[-]	Menge der natürlichen Zahlen
n_w	[-]	Anzahl der Verbindungsgewichte eines Neuronalen Netzes
NF	[-]	Netzübertragungsfunktion
$N(\mu, \sigma^2)$	[-]	Normalverteilung mit Mittelwert μ und Standardabweichung σ
\mathbf{o}	[-]	Vektor der Istausgabe
p	[-]	Anzahl der Muster eines Trainingsdatensystems
P	[N]	Äußere Last eines Fachwerks
\mathbb{P}	[-]	Gesamtparameterbereich
P_s	[-]	Zuverlässigkeit
P_f	[-]	Versagenswahrscheinlichkeit
$P(A)$	[-]	Wahrscheinlichkeit des Ereignisses A

\mathbb{R}	[-]	Menge der reellen Zahlen
$\text{Rg}(\mathbf{X})$	[-]	Rang der Koeffizientenmatrix \mathbf{X}
S	[-]	Überlebensbereich des Ereignisraums
$S_{j\ k}$	[N]	Elastizität bzw. Steifigkeit des Stabs $\text{Stab}_{j\ k}$
smod	[-]	Stochastisches Modell
SSE	[-]	<i>Sum Squared Error</i> (siehe Gleichung (85))
$\text{Stab}_{j\ k}$	[-]	Fachwerksstab zwischen den Knoten j und k
$\sup f$	[-]	Supremum der Funktion f
t	[-]	Vektor der Sollausgabe
T	[-]	Datensystem (i.A. Test- und Trainingsdaten)
T_R	[-]	Rosenblatt-Transformation
u	[mm]	Vektor der Knotenverschiebungen eines Fachwerks
v_{limit}	[mm]	Grenzauslenkung zur Bestimmung von P_f
V	[-]	Versagensbereich des Ereignisraumes
$\text{Var}[X]$	[-]	Varianz
$w_{i\ j}$	[-]	Verbindungsgewicht zwischen Neuron i und Neuron j
W	[-]	Matrix der Verbindungsgewichte
x	[-]	Punkt im Vektorraum \mathbb{R}^n
X	[-]	Koeffizientenmatrix
α_i	[1]	Struktureller Integritätskoeffizient
β	[-]	Zuverlässigkeitsindex
Γ_r	[1]	Dimensionslose Normalspannung des Stabs r
$\delta_{\text{Kn } i,y}$	[mm]	Auslenkung des Knotens i in y -Richtung
$\Delta_{i,y}$	[1]	Dimensionslose Auslenkung des Knotens i in y -Richtung
ΔE	[-]	Taylorreihenentwicklung der Gesamtfehlerfunktion E
θ	[-]	Schwellenwert
λ	[-]	Anzahl der Nachkommen einer Generation

μ_{test}	[%]	Mittlerer Testfehler eines Neuronalen Netzes
μ	[-]	Anzahl der Eltern einer Generation
μ	[-]	Erwartungswert der Normalverteilung
Π	[1]	Dimensionsloses Potenzprodukt
$(\Pi_1, \Pi_2, \dots, \Pi_p)$	[1]	Fundamentalsystem
ρ	$\left[\frac{\text{kg}}{\text{mm}^3} \right]$	Materialdichte Al-Legierung 3.4354
$\sigma_{j k}$	$\left[\frac{\text{N}}{\text{mm}^2} \right]$	Normalspannung des Stabs Stab _{j k}
σ	[-]	Standardabweichung der Normalverteilung
Φ	[-]	Standardnormalverteilung N(0,1)
Ψ	[1]	Verhältnis charakteristischer Stablängen
ANT	[-]	Ausgangsnetztopologie
NT	[-]	Netztopologie
FW	[-]	Fachwerk
[a, b]	[-]	abgeschlossenes Intervall
]a, b[[-]	offenes Intervall
$\{1, \dots, n\}$	[-]	Zahlenmenge
(1, 2, 3)	[-]	Vektor mit Koeffizienten 1, 2 und 3
$ \mathbf{x} $	[-]	Betrag von \mathbf{x}
$\ \mathbf{x}\ $	[-]	Norm von \mathbf{x} (z.B. euklidische Norm)
\mathcal{O}	[-]	Landau-Symbol für Funktionenklassen

1 Einleitung

Die Fähigkeiten und Potentiale des Gehirns und seines Grundbausteins Nervenzelle bei der Informationsverarbeitung motivieren die Untersuchung künstlicher Neuronaler Netze. Hierbei wird versucht, die dem biologischen Vorbild zugrundeliegenden komplexen elektrochemischen Vorgänge und ihre Interaktion innerhalb der massiv parallelen Netzstrukturen des Gehirns durch stark vereinfachte Grundkonzepte einer Nutzung in der Informationsverarbeitung zugänglich zu machen. Der Begriff Neuronale Netze umfaßt die Simulation künstlicher Neuronaler Netze mit Hilfe spezieller Computerprogramme. Neuronale Netze sind massiv parallele, lernfähige Datenverarbeitungssysteme, die in ihren Grundprinzipien nach Aufbau, Struktur, Netzplan und Interaktionen z.B. den Gehirnen höherer Lebewesen nachgeahmt sind. Die Lernfähigkeit künstlicher Neuronaler Netze begründet sich aus dem Potential, grundlegende Gesetzmäßigkeiten des in den Trainingsdaten beschriebenen Systemverhaltens als verteilte Wissensrepräsentation in den variablen Parametern des Netzwerks zu abstrahieren.

Die aus der natürlichen Evolution abgeleitete Evolutionsstrategie ist ein stochastisches Optimierungsverfahren. Sie versucht, die Funktionsprinzipien ausgewählter Mechanismen der biologischen Evolution für die Optimierung mathematisch-technischer Problemstellungen zu nutzen. Dieses Verfahren erlaubt die Untersuchung multimodaler Optimierungsaufgaben, ohne daß Vorkenntnisse zur untersuchten Zielfunktion notwendig sind. Grund ist der auf einer Entwicklungshistorie aufbauende, stochastische Charakter des Verfahrens sowie die selbstadaptive Steuerung der Strategie, angepaßt an die lokale Topologie der Zielfunktion. Evolutionsstrategien sind keine universellen Optimierungsverfahren. Grundsätzlich besteht die Möglichkeit, das gesuchte globale Extremum einer multimodalen Optimierungsaufgabe durch Evolutionsstrategien zu lokalisieren. Dies kann jedoch nicht mit Sicherheit vorhergesagt werden.

Im folgenden werden die Aufgabenstellungen der Arbeit und der aktuelle Stand der Forschung in den bearbeiteten Fachgebieten dargestellt.

1.1 Aufgabenstellung

Die Untersuchung der bei technischen Fragestellungen auftretenden, multimodalen Optimierungsaufgaben mehrerer, gleichzeitig zu berücksichtigender Optimierungskriterien ist in einem mehrdimensionalen Parameterraum aufgrund der dazu notwendigen Anzahl an Rechenoperationen selbst bei einfachen Aufgabenstellungen äußerst aufwendig. Im Rahmen der Arbeit wird eine Systematik entwickelt, die aufwendige und häufig auftretende Rechenope-

rationen (z.B. Restriktionskontrollen, Zielfunktionsbestimmungen) für definierte Parameterbereiche relevanter Optimierungskriterien und -restriktionen durch trainierte Neuronale Netze ersetzt. Gleichzeitig wird das Potential Neuronaler Netze für ein Verfahren zur Schadensüberwachung und -lokalisierung von Tragwerkssystemen untersucht. Als Beispiel sei die Schadensüberwachung von Flugzeugbauteilen im Betrieb genannt. In einem ersten Schritt wird ein Verfahren zur Analyse verschiedener Schädigungszustände statisch bestimmter und unbestimmter Fachwerkstrukturen nach Existenz, Ort und Ausmaß von Schädigungen entwickelt. Das hierzu grundlegende Verständnis der Korrelation Neuronaler Netze mit den Steifigkeiten der untersuchten Strukturen ist zu erarbeiten. Ziel ist die Integration geeigneter Neuronaler Netze in Systeme zur Optimierung und Überwachung technischer Strukturen. Die Untersuchungen der vorliegenden Arbeit untergliedern sich in die drei im folgenden beschriebenen Aufgabenstellungen.

Neuronale Netze werden zur effizienten Erfassung von Kriterien und Restriktionen der zu untersuchenden Optimierungsaufgabe unter Ausnutzung neuer Erkenntnisse aus dem Bereich lernfähiger informationsverarbeitender Systeme (z.B. Reduktion von Neuronalen Netzen durch Ausdünnungsverfahren sog. *Pruning*) eingesetzt. Ziel der Untersuchungen ist, Neuronale Netze über eine neue Systematik in das stochastische Optimierungsverfahren „Evolutionstrategie“ einzubinden und für Optimierungsaufgaben der Strukturmechanik zu testen. In Abbildung 20 ist das zu dieser Problemstellung entwickelte Schema der Einbindung trainierter Neuronaler Netze in Evolutionsstrategien zur Untersuchung von Optimierungsaufgaben mit mehreren Optimierungszielen dargestellt.

Weiterhin kann das Lernpotential Neuronaler Netze dazu genutzt werden, Rückschlüsse auf den Schädigungszustand einer statisch belasteten Struktur zu ziehen. Die Problemstellung entspricht mathematisch der Ermittlung eines unbekanntem inversen Zusammenhangs. Zur besseren praktischen Anwendbarkeit enthalten die Trainingsdaten einfach meßbare Größen wie Systemreaktionen. Ziel ist, Aussagen zu Existenz, Ort und Ausmaß einzelner oder mehrerer gleichzeitig auftretender Schädigungen aller Komponenten des untersuchten Tragwerks treffen zu können. Hierauf aufbauend sind Echtzeit-Überwachungssysteme auf Basis trainierter Neuronaler Netze oder entsprechender VLSI-Schaltungen denkbar, die signifikante Veränderungen im Spektrum der Systemreaktionen einer belasteten Struktur den jeweils aufgetretenen Schädigungen als Ursache zuordnen können.

Untersuchungen zu Korrelationen zwischen Steifigkeitsmatrizen von Fachwerkstrukturen und den Topologien zugehöriger Neuronaler Netze dienen einem verbesserten Verständnis der Einsatzmöglichkeiten Neuronaler Netze in der Strukturmechanik. Das Verschiebungsverhalten statisch belasteter Fachwerkstrukturen kann nach Theorie I. Ordnung durch ein inhomoge-

nes lineares Gleichungssystem (iLGS) beschrieben werden. Grundidee ist die Lösung dieses iLGS durch ein geeignetes, noch zu entwickelndes Neuronales Netz. Hiermit soll die Frage beantwortet werden, welche Neuronen oder Teilbereiche eines erfolgreich trainierten Neuronalen Netzes welche Teilbereiche der Steifigkeitsmatrix des hiermit beschriebenen statisch belasteten Fachwerks repräsentieren. Diese Zusammenhänge ermöglichen die zielgerichtete Auswahl derjenigen Neuronalen Netze, die am besten für die verschiedenen Teilaufgaben der zu untersuchenden Optimierung geeignet sind.

Die Resultate aller drei Teilaufgaben der Arbeit zusammengenommen erlauben die erfolgreiche Untersuchung von Optimierungsaufgaben mit mehreren Optimierungszielen im Rahmen der entwickelten Systematik (siehe Abbildung 20). Gleichzeitig wird damit die Funktionsfähigkeit und Effizienz der vorgeschlagenen Systematik nachgewiesen.

1.2 Stand der Forschung

Neuronale Netze reagieren auf die sie stimulierende Informationsumgebung durch Veränderung ihrer variablen Parameter. Für diese Lernfähigkeit sind der Netzplan der Neuronen, die Gewichtung der Verbindungen zwischen den Neuronen, die Informationsverarbeitung in den Neuronen (Neuronenfunktionalität) sowie die Leistungsfähigkeit der Lernalgorithmen beim Training der Neuronalen Netze ausschlaggebend. Die Anpassung der Freiheitsgrade an die untersuchte Problemstellung während des Trainings der Neuronalen Netze begründet ihre kognitive Leistungsfähigkeit.

Die folgenden Abschnitte bieten einen Überblick über den Stand der Forschung in den für die vorliegende Arbeit signifikanten Forschungsgebieten.

1.2.1 Neuronale Netze in der Strukturmechanik

Erste Überlegungen und Versuche zur Anwendung Neuronaler Netze in der Strukturmechanik sind bei MCAULEY [1], VANLUCHENE UND SUN [2] sowie REHAK ET AL. [3] zu finden.

SWIFT UND BATILL [4] untersuchen Neuronale Netze zur Vorauslegung von Strukturen über direkte Abbildung signifikanter Systemparameter auf Kenngrößen zu optimierender Zielfunktionen wie Masse, Knotenverschiebungen und Eigenfrequenzen am Beispiel von Fachwerkstrukturen und einem Flügelkasten.

PAO [5] verwendet in dem sog. *functional link concept* Funktionsansätze und bekannte Zusammenhänge zwischen Variablen der Eingabeschicht Neuronaler Netze als zusätzliche Ein-

gabegrößen und verbessert damit den Lernfortschritt bei gleichzeitiger Reduktion des Trainingsaufwands.

GUNARATNAM UND GERO [6] untersuchen Möglichkeiten, die Effektivität und Generalisierungsfähigkeit Neuronaler Netze durch Vorverarbeitung der Trainingsdaten mittels Dimensionsanalyse zu verbessern. Anwendungsbeispiele sind Problemstellungen aus dem Bereich Strukturanalyse.

HAJELA UND SZEWCZYK [7] leiten aus der Analyse der Verbindungsgewichte eines trainierten *Feedforward*-Netzwerks eine Übergangsmatrix zur Erfassung der Korrelation zwischen den Neuronen der Eingabe- und Ausgabeschicht ab. Analytische Testfunktionen und Beispiele aus der Strukturanalyse zeigen die Anwendbarkeit dieser Vorgehensweise zur Bestimmung der minimalen Anzahl notwendiger Trainingsdaten sowie zur Festlegung der Anzahl verdeckter Neuronen zugehörig zur jeweils untersuchten Problemstellung.

Erste Arbeiten zur Kopplung von Vorgehensweisen der Dimensionsanalyse nach dem Π -Theorem mit Ansätzen zur Entwicklung problemspezifischer Neuronaler Netztopologien für die analytisch exakte Erfassung implizit in Trainingsdaten enthaltener Zusammenhänge finden sich bei BARTH [8], EMRICH [9], HERRMANN [10], RUDOLPH [11, 12, 13, 14, 15] sowie RUDOLPH UND KRÖPLIN [16]. Hier wird auch eine ausführliche Darstellung von Anwendungsmöglichkeiten der Dimensionsanalyse u.a. in der Strukturmechanik gegeben. Dies ist in ihren Grundlagen für die vorliegende Arbeit von Bedeutung.

RUDOLPH [17] untersucht Möglichkeiten, über einen evolutionären Entwicklungsprozess durch Genetische Algorithmen optimal generalisierende Neuronale Netze zur Erfassung dimensions homogener physikalischer Vorgänge zu erzeugen. Genetische Algorithmen werden dabei zur Entwicklung und Auswahl von geeigneten Netztopologien eingesetzt. Zur selektiven Bewertung der Netztopologien wird eine speziell entwickelte Fitnessfunktion eingesetzt.

Aus den Arbeiten wird deutlich, welche Bedeutung der Dimensionsanalyse zur Datenvorverarbeitung sowie der Berücksichtigung a priori vorhandener Informationen zur Problemstellung bei der Entwicklung problemspezifischer Netztopologien und damit bei der exakten Erfassung unbekannter Zusammenhänge aus Datenmustern zukommt.

1.2.2 Neuronale Netze zur Zielfunktionsbeschreibung in der Optimierung

Der folgende Abschnitt gibt einen Einblick in Arbeiten zum Thema Kopplung Neuronaler Netze mit einem Optimierungsverfahren. Ziel ist die Untersuchung von Optimierungsaufga-

ben mit mehreren Optimierungszielen.

HAJELA UND BERKE [18] sowie BERKE UND HAJELA [19, 20] untersuchen Möglichkeiten, Systemparameter belasteter Fachwerke über Neuronale Netze auf zugehörige Systemantworten wie z.B. Weggrößen abzubilden. Trainierte Neuronale Netze dieser Art werden zusammen mit deterministischen Optimierungsverfahren zur Tragwerksoptimierung verwendet. Der Einsatz Neuronaler Netze als Speicher von Optimierungs- und Rechenergebnissen, bei dem trainierte Neuronale Netze gegebene Konstruktions- und Optimierungsbedingungen direkt mit zugehörigen Optimalstrukturen verknüpfen, ist ein weiterer verfolgter Ansatz.

HAJELA UND BERKE [21] entkoppeln multimodale Tragwerksoptimierungsprobleme in einzelne, durch Neuronale Netze gekoppelte und erfaßte Teilprobleme.

SZEWczyk UND HAJELA [22] untersuchen die Anwendbarkeit von *Counterpropagation*-Netzen (siehe z.B. ZELL [23]) zur Approximation von Stabspannungen in Abhängigkeit von Stabquerschnittsflächen in Verbindung mit *Simulated Annealing*-Verfahren zur Optimierung der untersuchten Strukturen. Die dabei erzielbare Approximationsgenauigkeit ist im allgemeinen ausreichend, Fachwerke mit fast optimalen Eigenschaften zu ermitteln.

SZEWczyk UND HAJELA [24] untersuchen aufbauend auf [7, 22] die Aufteilung nicht-hierarchischer Optimierungsaufgaben in autonome Untersysteme, die über globale Parameter und Strafterme gekoppelt werden. Neuronale Netze bilden Systemparameter auf zugehörige Systemreaktionen ab und dienen über eine Analyse der zugehörigen Übergangsmatrix der Erfassung von Korrelationen zwischen den Neuronen der Ein- und Ausgabeschicht. Anwendungsbeispiele sind ebene und räumliche Fachwerkstrukturen.

CARPENTER UND BARTHELEMY [25] vergleichen die Approximationsfähigkeit konventioneller Neuronaler Netze mit Polynomansätzen höherer Ordnung.

1.2.3 Schadensanalyse in Fachwerkstrukturen

Die Analyse und Vorhersage von Schädigungen von Tragwerken ist Gegenstand umfangreicher Forschung. Die folgende Zusammenfassung gibt einen Überblick über Forschungsarbeiten mit Relevanz zu Vorgehensweisen und Ergebnissen des Kapitels 8.

Der Schadensanalyse liegt als Grundaufgabe die Bestimmung eines unbekanntem inversen Zusammenhangs zwischen Ein- und Ausgabewerten zugrunde. Theoretische Untersuchungen zur eindeutigen Bestimmbarkeit des inversen Zusammenhangs finden sich im Hinblick auf nichtinvasive Methoden (siehe Abschnitt 8.1) bei SYLVESTER UND UHLMANN [26].

KAWIECKI [27] benutzt *Feedforward*-Netzwerke zur Vorhersage von Ort und Größe eines

Risses in Kragträgern. Das zur Generierung von Trainingsdaten verwendete Balkenmodell simuliert einen Riss als Drehfeder und arbeitet mit Biegefrequenzen sowie Amplitudeninformationen unter stationärer dynamischer Anregung.

TSOU UND HERMAN SHEN [28] arbeiten mit drei gekoppelten *Feedforward*-Netzwerken zur Datenvorverarbeitung sowie zur Analyse von Schädigungen nach Ort und nach Ausmaß. Anwendungsbeispiele sind Feder-Dämpfer-Systeme mit drei und acht Massen.

KUDVA ET AL. [29] untersuchen versteifte Platten unter einachsiger Belastung mittels Neuronaler Netze in zwei Schritten auf Schädigungen: das erste Netz führt eine grobe Schadenslokalisierung durch, während das zweite Netz Ort und Ausmaß der Schädigung genauer bestimmt. Die Schädigungen sind dabei durch kreisförmige Löcher realisiert. Trainingsdaten werden mittels FEM-Rechnungen (Finite-Elemente-Methode) ermittelt und bestehen aus dem aktuellen Belastungsmuster als Eingabe sowie Ort und Ausmaß des Schadens als Sollausgabe.

ELKORDY ET AL. [30] untersuchen Möglichkeiten, Schädigungen einer dreidimensionalen Rahmenstruktur durch trainierte Neuronale Netze zu bestimmen. Die Schädigungen sind dabei durch Schwächung der Diagonalversteifungen definiert. Als Eingabedaten kommen Veränderungen von Vibrationsmustern infolge der definierten Schädigungszustände und als Sollausgabedaten Informationen zu Existenz bzw. Ort und Ausmaß der Schädigungen zum Einsatz. Darauf aufbauend schlagen die Autoren in [31] ein *Online*-Schadensüberwachungssystem für Strukturen vor.

RHIM UND LEE [32] untersuchen am Beispiel eines durch Delamination geschädigten Kragträgers Neuronale Netze zur Klassifikation verschiedener nach Ausmaß und Bereich der Schädigungen eingeteilter Schadensklassen. Eine Übertragungsfunktion beschreibt die untersuchte Struktur nach Anregung und Systemreaktion. Deren Parameter sind die Eingabe, die Schadensklassen sowie die Sollausgabe der Test- und Trainingsdaten.

WORDEN [33] und WORDEN ET AL. [34] ermitteln aus dem euklidischen Abstand zwischen Eingabe- und tatsächlichen Ausgabewerten eines speziellen, trainierten Neuronalen Netzes einen Kennwert (*novelty index*). Dieser unterscheidet, ob sich das dem Netzwerk präsentierte Datenmuster einer Systemübertragungsfunktion oder Reflektionen von *lamb*-Schwingungen signifikant von bisher präsentierten Datenmustern unterscheidet und so auf die Existenz einer Schädigung hindeutet. Die Methode ist nur in der Lage, Aussagen über Existenz oder Nicht-Existenz eines Schadens zu treffen.

WU ET AL. [35] verwenden diskrete Fourier-Spektren errechneter und gemessener Zeit-Beschleunigungsverläufe in Reaktion auf eine dynamische Anregung der Struktur als Eingabe und zugehörige normierte Schädigungszustände der Strukturkomponenten als Sollaus-

gabe zum Training Neuronaler Netze. Anwendungsbeispiel ist eine Rahmenstruktur, in der Schädigungen durch Steifigkeitsreduktion realisiert sind.

Eine Methodik der Umkehrung von Ursache und Wirkung zur Erzeugung von Test- und Trainingsmustern Neuronaler Netze anhand von Reaktionen des geschädigten Systems wird von HERRMANN in [10, 36] vorgeschlagen und durch HERRMANN UND STELLBRINK [37] sowie HERRMANN ET AL. [38] weiter untersucht. Die Entwicklung einer Systematik zur Bestimmung inverser Zusammenhänge basierend auf den Systemreaktionen der geschädigten Struktur ist in Kapitel 8 untersucht. Die Veröffentlichungen erster Teilergebnisse dort beschriebener Untersuchungen finden sich bei HERRMANN UND STRENG [39, 40].

Weiterhin sei auf den Konferenzband *Structural Health Monitoring* von CHANG [41] hingewiesen.

1.2.4 Korrelation Struktur – Neuronale Netze

Die Lösung inhomogener linearer Gleichungssysteme durch spezielle Neuronale Netze ist Gegenstand verschiedener Veröffentlichungen. Die für die vorliegende Arbeit relevanten Forschungsergebnisse werden hier zur Abgrenzung eigener Resultate inhaltlich zusammengefaßt.

CHAKRABORTY ET AL. [42] entwickeln ein Hopfield-Netzwerk (siehe z.B. ZELL [23]) zur Lösung von iLGS über die Minimierung einer Energiefunktion. Hiermit ist die Lösung von Gleichungssystemen mit bis zu 300 Unbekannten möglich.

FORBES UND MANSFIELD [43] implementieren durch eine spezielle Neuronenfunktionalität und einen speziellen Netzplan der Neuronen den Huang-Algorithmus zur Lösung von iLGS in ein Neuronales Netz. Die dabei beschriebene Methodik unterscheidet sich bzgl. entwickelter Netztopologie und eingesetztem Lernverfahren signifikant von der neuen Systematik der vorliegenden Arbeit (siehe Kapitel 9).

MARGARITIS ET AL. [44] überführen das zu lösende iLGS in eine Normalenform. Die über ein Hopfield-Netzwerk durchgeführte Minimierung einer zugehörigen Energiefunktion löst das iLGS im Sinne minimaler Fehlerquadrate.

POLYCARPOU UND IOANNOU [45] setzen einen orthogonalisierten Backpropagation-Algorithmus (siehe z.B. ZELL [23]) zusammen mit einer *Feedforward*-Netztopologie zur Lösung eines iLGS in einer Lernepoche im Sinne minimaler Fehlerquadrate ein.

REKANOS UND TSIBOUKIS [46] verwenden Hopfield-Netzwerke mit sigmoiden Transferfunktion der Neuronen zur Lösung von iLGS aus Finite-Elemente-Untersuchungen elektrostatischer Feldprobleme.

WANG [47] stellt die elektronische Realisation eines Hopfield-Netzwerks zur Lösung von iLGS in Echtzeit vor.

WANG UND MENDEL [48, 49] ermitteln zur Lösung von iLGS im Sinne minimaler Fehlerquadrate die pseudo-inverse Koeffizientenmatrix des Gleichungssystems. Dies geschieht mittels zweischichtiger, ebener Netztopologien über die Festlegung einer Schicht von Netzgewichten auf die Parameter der gegebenen Koeffizientenmatrix.

WANG UND MENDEL bestimmen in [50] mit identischem Vorgehen über dreidimensionale, aus n zweischichtigen, ebenen Netztopologien aufgebauten Netzwerken die Lösung von iLGS.

EMRICH [9] und HERRMANN [10] lösen iLGS durch Abbildung eines Produktansatzes in eine problemspezifische Netztopologie. Hierzu wird den Neuronen der Eingabeschicht und der verdeckten Schicht eine spezielle Funktionalität zugeordnet. Die Methodik ist aber auf positive Parameter der Koeffizientenmatrix und des Vektors der Absolutglieder beschränkt.

Gegenüber der in Kapitel 9 vorgeschlagenen neuen Methodik zur Lösung von iLGS mittels problemspezifischer Neuronaler Netze zeigen alle hier zitierten Arbeiten signifikante Unterschiede.

2 Mathematik

Der Begriff Optimierung beschreibt allgemein die Suche nach der besten oder zumindest einer verbesserten Lösung der untersuchten Aufgabe. Entscheidend ist die Existenz eines Kriteriums, das die Einstufung einer Lösung als besser oder gar optimal ermöglicht. Bei Optimierungsverfahren kann grundsätzlich zwischen experimenteller und mathematischer Optimierung unterschieden werden.

Eine experimentelle Optimierung technischer Systeme ist möglich, falls das System charakterisierende, physikalische Größen gemessen werden können. Demgegenüber hängt die Möglichkeit, eine Problemstellung mit Hilfe mathematischer Instrumente zu untersuchen, davon ab, inwieweit es gelingt, das betrachtete System mathematisch zu beschreiben. Daraus kann dann eine Zielfunktion als Optimierungskriterium abgeleitet werden.

2.1 Grundlagen der Optimierung

Ein Optimierungsmodell stellt den mathematischen Zusammenhang zwischen den Variablen des Systems und einem Funktionswert als Qualitäts- bzw. Beurteilungskriterium her. Die Bewertungsvorschrift wird im folgenden als Ziel- oder Qualitätsfunktion bezeichnet. Der Zustandspunkt des Systems im Parameterraum ist durch einen n -dimensionalen reellwertigen Vektor charakterisiert. Als Parameter gelten die Gesamtheit aller Werte, die Einfluß auf die Qualität der Zielfunktion f haben.

Zur Lösung einer Optimierungsaufgabe ist derjenige n -dimensionale Ortsvektor $\mathbf{x}_0 \in \mathbb{R}^n$ im Parameterraum gesucht, für den die Zielfunktion f , dargestellt durch ihre Hyperfläche $(\mathbf{x}, f(\mathbf{x})) \in \mathbb{R}^n \times \mathbb{R}$, einen Extremwert annimmt. Das Extremum stellt eine beste Lösung abhängig von getroffenen Annahmen und Vereinfachungen des vereinbarten Optimierungsmodells dar.

2.1.1 Definitionen und Vereinbarungen

Der Zustandspunkt eines Systems im n -dimensionalen Vektorraum \mathbb{R}^n über den reellen Zahlen \mathbb{R} ist durch

$$\mathbf{x} = (x_1, \dots, x_n)^T \quad (1)$$

gegeben. Das Optimierungsproblem ist definiert als die Aufgabe, zu einer reellwertigen Zielfunktion

$$f: \begin{cases} \mathbb{M} \subseteq \mathbb{D} \subseteq \mathbb{R}^n & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto q \end{cases} \quad \text{mit } \mathbb{M} \neq \emptyset \quad (2)$$

für n unabhängige Variablen innerhalb des zulässigen, abgeschlossenen Teilbereiches \mathbb{M} ihres Definitionsbereiches \mathbb{D} ihr Optimum als n -Tupel reeller Zahlen x_i zu ermitteln. Die Zielfunktion weist dazu jedem Zustandspunkt \mathbf{x} eindeutig einen Qualitätswert $q \in \mathbb{R}$ zu. Der natürliche Definitionsbereich \mathbb{D} bezeichnet dabei diejenige Teilmenge aus \mathbb{R}^n , für die die Zielfunktion definiert ist. Restriktionen, die in Abschnitt 2.1.2 besprochen werden, schränken den Definitionsbereich auf einen zulässigen Bereich \mathbb{M} ein. Die Parameter der in der vorliegenden Arbeit untersuchten Optimierungsaufgaben sind kontinuierlich.

Ein Minimierungsproblem über einem zulässigen Bereich $\mathbb{M} \neq \emptyset$ heißt zulässig, wenn die Zielfunktion $f(\mathbf{x})$ über \mathbb{M} beschränkt ist und ihr Minimum über \mathbb{M} annimmt. Wenn $\mathbb{M} \neq \emptyset$ und kompakt sowie f stetig ist, existiert mindestens eine Lösung.

Der Wert $f(\mathbf{x}_{gM})$ der Funktion f wird als globales Minimum bezeichnet, wenn die Bedingung

$$f(\mathbf{x}_{gM}) \leq f(\mathbf{x}) \quad \text{für alle } \mathbf{x} \in \mathbb{M} : \quad (3)$$

erfüllt ist. Die Beschränktheit einer Funktion bedingt die Existenz einer Zahl q_{\min} bzw. q_{\max} , so daß

$$q_{\min} \leq f(\mathbf{x}) \quad \text{bzw.} \quad q_{\max} \geq f(\mathbf{x}) \quad \text{für alle } \mathbf{x} \in \mathbb{M} \quad (4)$$

gilt. Die kleinstmögliche Zahl q_{\min} wird als Infimum $\inf \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{M}\}$ bezeichnet und entspricht dem globalen Minimum $\min \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{M} \subseteq \mathbb{R}^n\}$ mit $q_{\min} \in \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{M}\}$. Die im

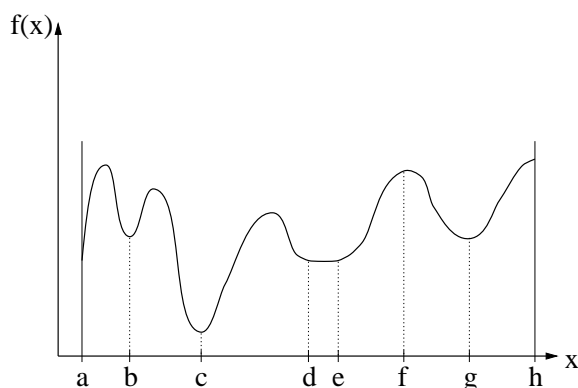


Abbildung 1: Schema einer Zielfunktion mit mehreren Extrema

Rahmen der vorliegenden Arbeit untersuchten Aufgabenstellungen sind sämtlich als Mini-

mierungsprobleme formuliert, was aber wegen

$$\inf_{\mathbf{x} \in \mathbb{M}} (f(\mathbf{x})) = - \sup_{\mathbf{x} \in \mathbb{M}} (-f(\mathbf{x})) \quad (5)$$

mit Supremum $\sup \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{M}\}$ keine Beschränkung der Allgemeinheit bedeutet. Unter diesen Bedingungen kann das globale Optimierungsproblem als Suche nach einem

$$\mathbf{x}_{gM} \in \mathbb{M} \text{ mit } f(\mathbf{x}_{gM}) = \min\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{M} \subseteq \mathbb{R}^n\} \quad (6)$$

angegeben werden. Der Wert $f(\mathbf{x}_{IM})$ wird als lokales Minimum von f bezeichnet, falls eine ε -Umgebung $U_\varepsilon(\mathbf{x}_{IM})$ von \mathbf{x}_{IM} existiert, so daß die Bedingung

$$f(\mathbf{x}_{IM}) \leq f(\mathbf{x}) \text{ für alle } \mathbf{x} \in \mathbb{M} \cap U_\varepsilon(\mathbf{x}_{IM}) \quad (7)$$

erfüllt ist. Damit muß jedes globale Minimum gleichzeitig auch lokales Minimum sein.

Eine Funktion $f : \mathbb{M} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ mit $\mathbb{M} \neq \emptyset$ heißt unimodal, wenn sie genau einen lokalen Minimalwert besitzt und die Menge der lokalen Minimalstellen zusammenhängend ist. Eine Funktion heißt multimodal, wenn sie nicht unimodal ist (GÖPFERT ET AL. [51]). Abbildung 1 zeigt eine eindimensionale Funktion $f(x)$ im abgeschlossenen Intervall $\mathbb{M} = \{x \mid a \leq x \leq h\}$. Lokale Minimalstellen der Funktion $f(x)$ befinden sich bei b, c, g und im Intervall $d \leq x \leq e$, wobei c auch globales Minimum ist. Die Funktion $f(x)$ ist im Intervall $a \leq x \leq h$ multimodal, dagegen z.B. in den Intervallen $d \leq x \leq f$ und $f \leq x \leq h$ unimodal.

2.1.2 Restriktionen

Die Parameter eines technischen Systems können normalerweise nicht beliebig gewählt werden. Häufig sind die Parameter aus technischen oder physikalischen Gründen verschiedenen Beschränkungen unterworfen und teilweise untereinander korreliert. Zum einen wird dem durch zulässige Wertebereiche der einzelnen Parameter Rechnung getragen. Zum anderen erfassen Neben- oder Restriktionsbedingungen über funktionale Zusammenhänge die Beschränkungen der Parameter.

Eine Einteilung nach Restriktionsumfang unterscheidet zwischen restriktionsfreien, quasi-restriktionsfreien und restriktionsbehafteten Optimierungsproblemen. Quasirestriktionsfreie Optimierungsprobleme sind nur durch obere und untere Grenzen für die Parameter beschränkt. Man spricht hierbei von Nebenrestriktionen. Bei restriktionsbehafteten Optimierungsproblemen wird jedem Parameter ein eigener zulässiger, abgeschlossener Wertebereich \mathbb{P}_k zugeordnet. Der zulässige Parameterbereich des Gesamtsystems $\mathbb{P} \subseteq \mathbb{R}^n$ ($\dim \mathbb{P}_k = 1$) ergibt sich als Produktraum aller zulässigen Einzelbereiche der Parameter zu

$$\mathbb{P} = \mathbb{P}_1 \times \mathbb{P}_2 \times \dots \times \mathbb{P}_n . \quad (8)$$

Das gleiche gilt für den natürlichen Definitionsbereich der Zielfunktion

$$\mathbb{D}_f = \mathbb{D}_{f,1} \times \mathbb{D}_{f,2} \times \dots \times \mathbb{D}_{f,n} . \quad (9)$$

Zusätzlich muß $\mathbb{P}_k \subseteq \mathbb{D}_{f,k}$ erfüllt sein.

Der Gesamtparameterbereich \mathbb{P} kann durch Beziehungen der Parameter untereinander (Nebenbedingungen) weiter auf den zulässigen Bereich \mathbb{M} beschränkt werden. Die reellwertigen funktionalen Zusammenhänge werden durch Gleichheitsrestriktionen der Form

$$g_j(\mathbf{x}) = 0 \text{ für alle } \mathbf{x} \in \mathbb{M} \text{ mit } j = 1, \dots, e \quad (10)$$

$$\text{mit } g_j : \begin{cases} \mathbb{D}_{g_j} \subseteq \mathbb{R}^n & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto g_j(\mathbf{x}) \end{cases} \quad (11)$$

und Ungleichheitsrestriktionen der Form

$$g_i(\mathbf{x}) \leq 0 \text{ für alle } \mathbf{x} \in \mathbb{M} \text{ mit } i = (e+1), \dots, m \quad (12)$$

$$\text{mit } g_i : \begin{cases} \mathbb{D}_{g_i} \subseteq \mathbb{R}^n & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto g_i(\mathbf{x}) \end{cases} \quad (13)$$

berücksichtigt. Jede Gleichheitsrestriktion reduziert die Dimension n des Parameterraums im allgemeinen um eins. Die Menge \mathbb{M} der zulässigen Zustandspunkte des Systems ergibt sich als die Teilmenge von \mathbb{P} , für die alle Nebenbedingungen erfüllt sind. Jede Gleichheitsrestriktion kann durch zwei Ungleichheitsrestriktionen ersetzt werden. Die Ungleichheitsrestriktion $g_i(\mathbf{x}) \leq 0$ heißt im Punkt $\mathbf{v} \in \mathbb{P}$

$$\begin{aligned} & \text{verletzt, falls } g_i(\mathbf{v}) > 0, \\ & \text{erfüllt, falls } g_i(\mathbf{v}) \leq 0, \\ & \text{aktiv erfüllt, falls } g_i(\mathbf{v}) = 0 \\ & \text{und inaktiv erfüllt, falls } g_i(\mathbf{v}) < 0 \end{aligned}$$

gilt. Da die globalen Extrema auch auf dem Rand des zulässigen Bereichs auftreten können, ist es sinnvoll, diesen in der Formulierung der Nebenbedingungen mit zu berücksichtigen.

Es ergibt sich mit den Gleichungen (2) und (10) bis (13) eine allgemeine Formulierung der restringierten Optimierungsaufgabe zu

$$\begin{aligned} \text{Problem } \mathcal{P}1 : \quad & f(\mathbf{x}) = \min! & \mathbf{x} & \in \mathbb{P} \subseteq \mathbb{D} \subseteq \mathbb{R}^n \\ & g_j(\mathbf{x}) = 0 & j \in \mathbb{J} & = \{1, \dots, e\} \\ & g_i(\mathbf{x}) \leq 0 & i \in \mathbb{I} & = \{(e+1), \dots, m\} . \end{aligned} \quad (14)$$

2.2 Optimierungsverfahren

Grundsätzlich ist bei mathematischen Optimierungsverfahren zwischen deterministischen und stochastischen Methoden zu unterscheiden. Bei stochastischen Verfahren beruhen bestimmte Entscheidungsklassen bewußt auf Zufallsprozessen. Über die Steuerung der Lageparameter der zugehörigen Wahrscheinlichkeitsverteilungen kann der Optimierungsfortschritt beeinflußt werden. Die Verfahren benötigen außer der Zielfunktion und den Randbedingungen keine weiteren Informationen über das Optimierungsproblem. Deterministische Verfahren sind auf spezielle Klassen von Optimierungsproblemen ausgelegt und damit nicht allgemein einsetzbar. Sie verwenden Informationen über die Zielfunktion wie z.B. Ableitungen und haben Stetigkeitsanforderungen (z.B. Lipschitz-Kriterium).

Im folgenden werden beispielhaft einige deterministische und stochastische Optimierungsverfahren genannt. Eine Übersicht geben z.B. SCHWEFEL [52] und SCHITTKOWSKI ET AL. [53] für Tragwerksoptimierungsverfahren, ARORA ET AL. [54] für globale Optimierungsmethoden und SPELLUCCI [55] für allgemeine mathematische Verfahren der nichtlinearen Optimierung.

2.2.1 Überblick über deterministische Verfahren

Als deterministisch gelten in der vorliegenden Arbeit alle Optimierungsverfahren, die in keiner Form Zufallsprozesse einsetzen. Verfahren der unrestringierten Optimierung setzen sich im allgemeinen aus Methoden zur Bestimmung der Suchrichtung und zugehöriger Schrittweite zusammen. Als Beispiel sei das Quasi-Newton-Verfahren genannt. Für restringierte und im allgemeinen nichtlineare Optimierungsprobleme unterscheidet man nach primalen Verfahren, Strafverfahren, dualen Verfahren und Approximationsmethoden (SPELLUCCI [55]).

Primale Verfahren erzeugen eine im Sinne des globalen Minimums äquivalente Ersatzziel-funktion. Das gleichungsrestringierte Problem ist durch die im Iterationspunkt aktiven Randbedingungen gegeben und wird zu jedem Iterationsschritt gelöst (GÖPFERT ET AL. [51]). Als Beispiel seien Projektions- und Reduktionsverfahren genannt.

Bei der Straffunktionsmethode unterscheidet man nach äußerer, innerer und erweiterter Methode. Allgemein wird dazu das Originalproblem in eine quasirestriktionsfreie oder eine restriktionsfreie Formulierung umgesetzt. Die Ersatzzielfunktion setzt sich additiv aus der Zielfunktion und einem aus den aktiven Restriktionen aufgebauten Strafterm zusammen. Die lokalen bzw. globalen Lösungen der Ersatzformulierung können mit Verfahren der unrestringierten Optimierung bestimmt werden und nähern die Lösung der restringierten Original-

aufgabe an. Eine Annäherung an die Grenzen des zulässigen Gebiets bei der inneren Straffunktion oder ein Verlassen des zulässigen Gebiets bei der äußeren oder der ausgedehnten inneren Straffunktion wird mit einer Erhöhung des Zielfunktionswerts bestraft.

Bei den dualen Methoden (siehe z.B. SPELUCCI [55]) lassen sich unter Regularitätsvoraussetzungen die nichtlinearen und mit Ungleichungen beschränkten Problemstellungen über eine erweiterte Lagrange-Funktion lokal in quasi unrestringierte bzw. nur vorzeichenrestringierte Optimierungsaufgaben transformieren. Deren lokale Minimalstellen stimmen mit den lokalen Lösungen der ursprünglichen Aufgabe überein (z.B. Multiplikatormethode von Rockafellar).

2.2.2 Überblick über stochastische Verfahren

Die einfachste Methode ist die reine Zufallssuche (Monte-Carlo-Methode) ohne Berücksichtigung der Historie des Optimierungsverlaufs. Die Zielfunktion wird in zufällig gleichverteilt ausgewählten Zustandspunkten des Optimierungsraums untersucht und die ermittelten Funktionswerte mit dem bisher besten Funktionswert verglichen.

Bei der *Multistart Method* dient jeder zufällig ermittelte Punkt als Startwert einer lokalen Minimumsuche. Die so ermittelten lokalen Minima werden gespeichert und verglichen. Mit zunehmender Anzahl untersuchter Punkte strebt die Wahrscheinlichkeit der Verfahren, einen Punkt in einer gewissen Umgebung des globalen Minimums zu finden, gegen eins. Beide Verfahren sind mit hohem Rechenaufwand verbunden.

Threshold Accepting oder der sogenannte Sintflut-Algorithmus gehen ähnlich dem *Simulated Annealing*-Verfahren (siehe z.B. VIDAL [56]) von einem zulässigen Startwert aus, der stochastisch leicht verändert wird. Ist die Differenz zwischen neuer und alter Qualität kleiner als ein positiver Schwellwert, so wird das Derivat neuer Startvektor; andernfalls wird ein neues Derivat generiert. Tritt während einer festgelegten Anzahl von Iterationen keine Qualitätsverbesserung ein, so wird der Betrag des Schwellwertes gesenkt. Dieses und daraus abgeleitete Verfahren zeigen sehr gute Ergebnisse z.B. beim *Traveling Salesman Problem* (DUECK [57] und DUECK UND SCHEUER [58]).

Evolutionäre Algorithmen umfassen alle Verfahren, die ausgewählte evolutionäre bzw. naturanaloge Mechanismen als Kernelemente einer mathematischen Optimierungsstrategie einsetzen. Als Beispiele seien hier Genetische Algorithmen (GA), Evolutionäre Programmierung (EP) und Evolutionsstrategien (ES) genannt. Die Methoden arbeiten mit einer Abfolge von Generationen, die iterativ den Optimierungsverlauf der in Populationen organisierten Individuen charakterisieren. Der zu jedem Individuum gehörende Variablenvektor kennzeichnet als mögliche Lösung der untersuchten Optimierungsaufgabe den aktuellen Aufenthaltsort des

Individuums im Optimierungsraum.

Evolutionsstrategien werden in Abschnitt 3.3 besprochen. Sie können als Abstraktion der biologischen Evolution auf der Ebene der Individuen verstanden werden.

2.3 Dimensionsanalyse

In dem Abschnitt werden grundlegende Begriffe und mathematische Zusammenhänge der Dimensionsanalyse, soweit sie für die in Abschnitt 6.1.1 beschriebene Datenvorverarbeitung von Relevanz sind, dargestellt. Detaillierte Ausführungen zum Thema Dimensionsanalyse finden sich z.B. bei GÖRTLER [59] und BRIDGMAN [60].

Die positiven Maßzahlen (Zahlenwerte) x_j von $j = 1, \dots, n$ physikalischen Größen haben zugehörig zum Grundgrößensystem $\{M_1, M_2, \dots, M_m\}$ die Dimension

$$[x_j] = M_1^{a_{j1}} \cdot M_2^{a_{j2}} \cdot \dots \cdot M_m^{a_{jm}} \quad (15)$$

als Dimensionsformel eines Potenzprodukts der Grundgrößen M_k . Die den a_{jk} zugehörige Matrix

$$\mathbf{A} := (a_{jk}) \quad \text{mit} \quad \begin{cases} j = 1, \dots, n \\ k = 1, \dots, m \end{cases} \quad (16)$$

heißt Dimensionsmatrix. Das Spaltenschema der Dimensionstafel in Tabelle 1 legt die Zuord-

A	M_1	M_2	\dots	M_m
x_1	a_{11}	a_{12}	\dots	a_{1m}
x_2	a_{21}	a_{22}	\dots	a_{2m}
\vdots	\dots	\dots	\dots	\dots
x_n	a_{n1}	a_{n2}	\dots	a_{nm}

Tabelle 1: Dimensionstafel der Dimensionsmatrix **A**

nung der Exponenten a_{jk} aus Gleichung (15) zu den Grundgrößen M_k fest. Im Zeilenschema kennzeichnen (x_1, x_2, \dots, x_n) die Maßzahlen der physikalischen Größen mit den zugehörigen Zeilenelementen a_{jk} als Exponenten der Dimensionsformel in Gleichung (15).

Eine reellwertige Funktion f wird als dimensionshomogen in den reellen Variablen x_j mit $j = 1, \dots, n$ bezeichnet, falls reelle Zahlen a_{jk} und b_k existieren, so daß

$$f(\alpha_1^{a_{11}} \cdot \dots \cdot \alpha_m^{a_{1m}} \cdot x_1, \dots, \alpha_1^{a_{n1}} \cdot \dots \cdot \alpha_m^{a_{nm}} \cdot x_n) = \alpha_1^{b_1} \cdot \dots \cdot \alpha_m^{b_m} f(x_1, \dots, x_n) \quad (17)$$

für alle $\alpha_k > 0$ mit $k = 1, \dots, m$ und x_j mit $j = 1, \dots, n$ im Definitionsbereich von f gilt (GÖRTLER [59]). Es besteht allgemein der Konsens, daß jeder richtig gewählte naturwissenschaftliche Modellansatz dem erkenntnistheoretischen Prinzip der Dimensionshomogenität gehorcht.

Ein physikalischer Zusammenhang mit hierauf basierendem Grundgrößensystem $\{M_1, M_2, \dots, M_m\}$ ist durch die positiven Maßzahlen x_j mit $j = 1, \dots, n$ von n physikalischen Größen beschreibbar. Dieses Grundgrößensystem ist nach erfolgter Wahl unverändert beizubehalten. Die Potenzprodukte

$$P_i := x_1^{k_{i1}} x_2^{k_{i2}} \cdot \dots \cdot x_n^{k_{in}} \quad (18)$$

sind gemäß Gleichung (15) mit $i = 1, \dots, p$ die Maßzahlen physikalischer Größen der Dimension

$$[P_i] = M_1^{p_{i1}} \cdot M_2^{p_{i2}} \cdot \dots \cdot M_m^{p_{im}} \quad (19)$$

Für

$$\mathbf{P} := (p_{ik}) := \sum_{j=1}^n k_{ij} a_{jk} \quad \text{mit} \quad \begin{cases} i = 1, \dots, p \\ k = 1, \dots, m \end{cases} \quad \text{und} \quad \mathbf{K} := (k_{ij}) \quad \text{mit} \quad \begin{cases} i = 1, \dots, p \\ j = 1, \dots, n \end{cases} \quad (20)$$

folgt in Matrixschreibweise die Gleichung $\mathbf{P} = \mathbf{K} \mathbf{A}$. Potenzprodukte der Form

$$\Pi_i := \phi_i(\mathbf{x}, \mathbf{k}_i) := x_1^{k_{i1}} x_2^{k_{i2}} \cdot \dots \cdot x_n^{k_{in}} \quad \text{mit} \quad i = 1, \dots, p \quad (21)$$

werden als dimensionslos bezeichnet, wenn für die physikalische Größe (Dimension) mit der Maßzahl Π_i

$$[\Pi_i] = M_1^0 \cdot M_2^0 \cdot \dots \cdot M_m^0 = 1 \quad (22)$$

gilt. Ein dimensionsloses Potenzprodukt gehorcht mit $\mathbf{P} = \mathbf{K} \mathbf{A}$ somit der Gleichung

$$(k_{i1}, k_{i2}, \dots, k_{in}) \mathbf{A} = \mathbf{0} \quad (23)$$

Das System $(\Pi_1, \Pi_2, \dots, \Pi_p)$ wird als Fundamentalsystem dimensionsloser Potenzprodukte bezeichnet, wenn es ein unabhängiges System von Potenzprodukten ist und wenn jedes beliebige dimensionslose Potenzprodukt Π_i der Maßzahlen (x_1, x_2, \dots, x_n) sich als Potenzprodukt der $(\Pi_1, \Pi_2, \dots, \Pi_p)$ darstellen läßt. Ein Fundamentalsystem dimensionsloser Potenzprodukte Π_i der Maßzahlen (x_1, x_2, \dots, x_n) setzt sich damit aus $p = n - r$ Potenzprodukten, wie in Gleichung (21) dargestellt, zusammen (GÖRTLER [59]). Hierbei bezeichnet r den Rang der Dimensionsmatrix \mathbf{A} . Die $i = 1, \dots, p$ Exponentenvektoren $(k_{i1}, k_{i2}, \dots, k_{in})$ stellen ein Fundamentalsystem von Lösungsvektoren $(\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_p)^T$ mit $\mathbf{k}_i = (k_{i1}, k_{i2}, \dots, k_{in})$ des homogenen Gleichungssystems

$$(\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_p)^T \mathbf{A} = \mathbf{0} \quad (24)$$

dar. Die zu einem beliebigen dimensionslosen Potenzprodukt Π_b gehörige Lösung $(k_{b1}, k_{b2}, \dots, k_{bn})$ ergibt sich als Linearkombination der p Lösungen eines Fundamentalsystems der Form

$$(k_{b1}, k_{b2}, \dots, k_{bn}) = \sum_{i=1}^p \lambda_i (k_{i1}, k_{i2}, \dots, k_{in}) \quad (25)$$

mit geeigneten reellen Faktoren $(\lambda_1, \lambda_2, \dots, \lambda_p)$.

Aus den grundsätzlichen Überlegungen folgt das für die Datenvorverarbeitung durch Dimensionsanalyse wichtige Π -Theorem in einer Formulierung nach GÖRTLER [59]. Es seien (x_1, x_2, \dots, x_n) Maßzahlen von n physikalischen Größen und \mathbf{A} nach Gleichung (16) die zugehörige Dimensionsmatrix in einem Grundgrößensystem $\{M_1, M_2, \dots, M_m\}$. Weiter sei f eine beliebige, dimensionshomogene Funktion in (x_1, x_2, \dots, x_n) auf dem Definitionsbereich $\mathbb{D} \subseteq \mathbb{R}_+^n$ sowie $(\Pi_1, \Pi_2, \dots, \Pi_p)$ mit $p = n - r$ und $r = \text{Rg}(\mathbf{A})$ ein beliebiges Fundamentalsystem dimensionsloser Potenzprodukte aus (x_1, x_2, \dots, x_n) . Dann existiert eine Funktion G von p Variablen und es existieren reelle Zahlen $\{k_1, k_2, \dots, k_n\}$ so daß für alle $P = \{x_1, x_2, \dots, x_n\} \in \mathbb{D}$ gilt:

$$f(x_1, x_2, \dots, x_n) = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n} G(\Pi_1, \Pi_2, \dots, \Pi_p) . \quad (26)$$

Ist ein physikalischer Zusammenhang f durch n Variablen x_j vollständig charakterisiert, so existiert eine Funktion G von $p \leq n$ dimensionslosen Potenzprodukten Π_i der x_j , die den zugrundeliegenden physikalischen Zusammenhang ebenfalls vollständig beschreibt. Das Π -Theorem ermöglicht die Reduktion der Anzahl der Problemvariablen um den Rang der Dimensionsmatrix. Im nichttrivialen Fall ($r > 0$) ist damit eine Reduktion der n Argumente der Funktion f auf $p = n - r$ Argumente der Funktion G möglich.

Die durch die Dimensionsanalyse maximal erreichbare Datenreduktion ist durch die Zahl der zum System gewählten Grundgrößen (z.B. SI-System) beschränkt. Prinzipiell besteht aber die Möglichkeit, durch die Einführung künstlicher, problemspezifischer Grundgrößen eine weitergehende Reduktion der Problemparameter zu erzielen (LANGHAAR [61]).

3 Tragwerksoptimierung

Die Tragwerksoptimierung verfolgt das Ziel, Tragwerke so zu gestalten, daß gegebene Qualitätskriterien bzw. Zielfunktionen unter Berücksichtigung von Restriktionen optimiert werden. Im Rahmen der vorliegenden Arbeit bezieht sich der Begriff Tragwerksoptimierung auf die Optimierung von Fachwerkstrukturen mit Nebenbedingungen. Die hierzu untersuchten Nebenbedingungen sind in Abschnitt 7.4.1 definiert.

Nach einleitenden Begriffsdefinitionen und Vereinbarungen folgt die Darstellung verschiedener Ersatzprobleme zur skalaren Erfassung von Optimierungsaufgaben mit mehreren Optimierungszielen. Anschließend wird die zur Tragwerksoptimierung eingesetzte mehrgliedrige Evolutionsstrategie besprochen und mit deterministischen Optimierungsverfahren verglichen.

3.1 Grundlagen

In diesem Abschnitt werden grundlegende Begriffe sowie Definitionen der Tragwerksoptimierung festgelegt. Eine umfassende Darstellung findet sich z.B. bei Petzold [62].

3.1.1 Strukturdefinitionen und Vereinbarungen

Eine Fachwerkstruktur ist durch die Anzahl der Knoten, Anzahl der Stäbe mit zugehöriger Verknüpfungsmatrix, Knotenkoordinaten, Definition der Lagerung sowie die Materialkennwerte und geometrischen Festlegungen definiert. Die Verknüpfungsmatrix legt die Systematik fest, nach der die Knoten durch Stäbe verbunden sind.

Das Problem der Optimierung von Fachwerkstrukturen ist in drei hierarchisch getrennte Klassen unterteilbar:

- **Topologieoptimierung:** Der übergeordnete Prozeß ermittelt eine optimale Topologie der Struktur unter Berücksichtigung der Randbedingungen. Die Topologie ist durch die Anzahl der Knoten mit zugehöriger Verknüpfungssystematik der Stäbe, die Definition der Lagerung und Belastung sowie durch geometrische Randbedingungen festgelegt (Topologievariablen).
- **Formoptimierung:** Die Optimierungsklasse legt die geometrischen Positionen der Knoten fest (Formvariablen).
- **Dimensionsoptimierung:** Nach Ermittlung einer zumindest lokal optimalen und zulässigen Topologie und der Festlegung geeigneter Knotenkoordinaten erfolgt die Optimie-

rung dieses Zwischenresultats hinsichtlich der Stabquerschnittsflächen (Dimensionsvariablen).

Topologievariablen, Formvariablen und Dimensionsvariablen zusammengenommen bilden die Menge der Strukturvariablen einer vollständigen Tragwerksoptimierungsaufgabe.

3.1.2 Optimierung

Die Tragwerksoptimierungsaufgabe ist als iterative Abfolge von Topologie-, Form- und Dimensionsoptimierung zu verstehen. Die Einzelaufgaben sind einander in der genannten Hierarchie untergeordnet.

Die Strukturparameter definieren die im Sinne der Optimierung konstanten Eigenschaften und Parameter einer Struktur. Hierzu zählen für den Fall einer Dimensionsoptimierung die Materialkenngrößen, Anzahl der Knoten, Lastfälle, Lagerbedingungen sowie Anzahl und Verknüpfung der Stäbe. Die Strukturvariablen sind die Veränderlichen der Optimierungsaufgabe und spannen den zugehörigen Suchraum auf. Im Fall der Dimensionsoptimierung umfaßt dieser Begriff die Stabquerschnitte der zu optimierenden Struktur. Strukturvariablen und Strukturparameter zusammengenommen definieren die in der Optimierungsaufgabe untersuchte Struktur vollständig.

Das Optimierungsmodell ist durch die Zielfunktion, die Restriktionsbedingungen, die Lastfälle sowie durch die Strukturparameter und Strukturvariablen definiert. Die Strukturvariablen sollten innerhalb des Optimierungsprozesses so bestimmt werden, daß die Optimierungsaufgabe für zulässige Werte der Randbedingungen bei gegebenen Strukturparametern bestmöglich erfüllt wird. Der Architektur aller Optimierungsverfahren ist eine Aufteilung in vier Hauptkomponenten gemein (z.B. Schmid [63]):

- Optimierungsmodell (Entwurfsaufgabe): Oberste Hierarchiestufe eines Optimierungsverfahrens. Definiert das Tragwerk und die die Optimierungsaufgabe definierenden Variablen.
- Analysemodell: Ermittelt das Verhalten der untersuchten Struktur z.B. als Reaktion auf eine Belastung. Die dazu eingesetzte Strukturanalyse (z.B. FE-Methode) liefert die Strukturantwort zugehörig zu den Strukturparametern, den Belastungen und den aktuellen Werten der Strukturvariablen (hier: Trainierte Neuronale Netze (siehe Kapitel 6)).
- Optimierungsalgorithmus (hier: Mehrgliedrige Evolutionsstrategie (siehe Abschnitt 3.3)): ein Element der Problemlösung. Der Optimierungsalgorithmus löst die durch

das Optimierungsmodell gestellte Optimierungsaufgabe und kommuniziert mit dem Analysemodell, um Daten zum untersuchten Tragwerk in der aktuellen Optimierungssituation zu erhalten.

- Kopplungsmodell: Verbindet Optimierungs- und Analysemodell sowie den Optimierungsalgorithmus (hier: Ersatzzielfunktionen (siehe Abschnitt 3.2)).

Die durchgeführten Untersuchungen zur Tragwerksoptimierung durch Einbindung Neuronaler Netze in Evolutionsstrategien werden am Beispiel der Dimensionsoptimierung statisch bestimmter Fachwerke durchgeführt. Dies ist detailliert in Kapitel 7 besprochen.

3.2 Ersatzzielfunktionen für Vektoroptimierungsprobleme

Technische Problemstellungen lassen sich vielfach nur als Optimierungsaufgabe mit mehreren, gleichzeitig zu berücksichtigenden Optimierungskriterien formulieren.

3.2.1 Vektoroptimierungsproblem

Seien \mathbb{R}^n bzw. \mathbb{R}^m n - bzw. m -dimensionale euklidische Vektorräume. Das System von m reellen Zielfunktionen f_j für $j = 1, \dots, m$ in n Veränderlichen $\mathbf{x} = (x_1, \dots, x_n)^T$

$$f_j = \begin{cases} \mathbb{D} \subset \mathbb{R}^n & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto y_j = f_j(\mathbf{x}) \end{cases} \quad (27)$$

mit gemeinsamem Definitionsbereich $\mathbb{D} \subset \mathbb{R}^n$ kann als Abbildung eines Teils von \mathbb{R}^n in einen Teil von \mathbb{R}^m in der Form

$$\mathbf{f} = \begin{cases} \mathbb{D} \subset \mathbb{R}^n & \rightarrow \mathbb{Y} \subset \mathbb{R}^m \\ \mathbf{x} & \mapsto (y_1, \dots, y_m) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix} \end{cases} \quad (28)$$

interpretiert werden. Damit wird jedem Vektor aus $\mathbf{x} \in \mathbb{D} \subset \mathbb{R}^n$ ein Vektor $\mathbf{y} \in \mathbb{Y} \subset \mathbb{R}^m$ eindeutig zugeordnet. Der Vektor $\mathbf{y} \in \mathbb{Y}$ kann dabei das Bild verschiedener $\mathbf{x} \in \mathbb{D}$ sein. Die Menge der Abbildungen f_j kann in der Form

$$F_m^n := \{\mathbf{f}: \mathbb{D} \subset \mathbb{R}^n \rightarrow \mathbb{Y} \subset \mathbb{R}^m\} \quad (29)$$

geschrieben werden.

Aus der simultanen Berücksichtigung mehrerer, i.a. miteinander konkurrierender Optimierungskriterien f_j ergibt sich die Mehrzieloptimierungsaufgabe (Vektoroptimierungsaufgabe) aufbauend auf der nichtlinearen, restringierten Optimierungsaufgabe $\mathcal{P}1$ in Gleichung (14). Für $\mathbf{f} \in F_m^n$ als stetige Abbildung nach Gleichung (28) auf dem durch die Restriktionen

$$\begin{aligned} g_k(\mathbf{x}) &= 0 & (g_k \in F_1^n; k = 1, \dots, e) \\ g_l(\mathbf{x}) &\leq 0 & (g_l \in F_1^n; l = (e+1), \dots, q) \end{aligned} \quad (30)$$

festgelegten Definitionsbereich $\mathbb{D} \subset \mathbb{R}^n$ ergibt sich die Vektoroptimierungsaufgabe zu

$$\mathbf{f}(\mathbf{x}) = \min! \quad \text{mit } \mathbf{x} \in \mathbb{D} . \quad (31)$$

Den Vektor $\mathbf{x}^* \in \mathbb{D}$ bezeichnet man als effizient bzgl. \mathbb{D} , falls kein Vektor $\mathbf{x} \in \mathbb{D}$ mit den Eigenschaften

$$\begin{aligned} x_i &\leq x_i^* & \text{für alle } i = 1, \dots, n \\ x_i &< x_i^* & \text{für mindestens ein } i \in \{1, \dots, n\} \end{aligned} \quad (32)$$

existiert. Die Effizienz wird als Pareto-Optimalität nach PARETO [64] bezeichnet. Die bzgl. \mathbb{D} effizienten Vektoren liegen stets auf dem Rand von \mathbb{D} . Ein Vektor $\mathbf{x}^* \in \mathbb{D}$ wird als funktional-effizient bzgl. \mathbb{D} und den Funktionen $\{f_1, \dots, f_m\}$ bezeichnet, falls kein Vektor $\mathbf{x} \in \mathbb{D}$ mit den Eigenschaften

$$\begin{aligned} f_j(\mathbf{x}) &\leq f_j(\mathbf{x}^*) & \text{für alle } j = 1, \dots, m \\ f_j(\mathbf{x}) &< f_j(\mathbf{x}^*) & \text{für mindestens ein } j \in \{1, \dots, m\} \end{aligned} \quad (33)$$

existiert. Die Menge

$$\mathbb{D}^* := \{\mathbf{x}^* \in \mathbb{D} : \mathbf{x}^* \text{ ist funktional-effizient bzgl. } \mathbb{D} \text{ und } \{f_1, \dots, f_m\}\} \quad (34)$$

heißt vollständige Lösungsmenge des Vektoroptimierungsproblems aus Gleichung (31). Damit kann die Minimierungsaufgabe des Vektoroptimierungsproblems aus Gleichung (31) als Bestimmung der vollständigen Lösungsmenge \mathbb{D}^* interpretiert werden.

3.2.2 Ersatzzielfunktion

Ez sei eine stetige Funktion aus F_1^n . Die Minimierungsaufgabe

$$\mathbf{Ez}[\mathbf{f}(\mathbf{x})] = \min! \quad \text{mit } \mathbf{x} \in \mathbb{D} \quad (35)$$

heißt genau dann Ersatzproblem für das Vektoroptimierungsproblem aus Gleichung (31), wenn ein $\tilde{\mathbf{x}} \in \mathbb{D}^*$ mit

$$\mathbf{Ez}[\mathbf{f}(\tilde{\mathbf{x}})] = \min \{ \mathbf{Ez}[\mathbf{f}(\mathbf{x})] \mid \mathbf{x} \in \mathbb{D} \} \quad (36)$$

existiert. Die Funktion \mathbf{Ez} wird dann als Ersatzzielfunktion bezeichnet und bewirkt eine Skalarisierung des Vektoroptimierungsproblems. Je nach Präferenz der Entscheidungsträger unterscheidet man für Ersatzzielfunktionen u.a. zwischen Abstandsfunktionen, einer Formulierung als MinMax-Problem und der Summation gewichteter Einzelzielfunktionen. Zu jeder der Funktionsklassen wird eine Ersatzzielfunktion vorgestellt. Um eine sinnvolle Gewichtung der Einzelzielfunktionen zur Ersatzzielfunktion zu ermöglichen, sind die Einzelzielfunktionen $f_j(\mathbf{x})$ auf ein gemeinsames Parameterintervall normiert.

Unter Berücksichtigung einer euklidischen Metrik ergibt sich als Abstandsfunktion für die erste Ersatzzielfunktion \mathbf{Ez}_1 über den zu minimierenden Abstand zwischen aktuellem Funktionswert $f_j(\mathbf{x})$ und Zielfunktionswert \bar{y}_j die Formulierung

$$\mathbf{Ez}_1[\bar{\mathbf{y}}] = \min! \quad (37)$$

$$g[\mathbf{f}(\mathbf{x}), \bar{\mathbf{y}}] = \sqrt{\sum_{j=1}^m w_j (f_j(\mathbf{x}) - \bar{y}_j)^2} \quad (38)$$

mit $0 \leq w_j \leq 1$ und $\sum_{j=1}^m w_j = 1$. Der Vektor $\bar{\mathbf{y}} = (\bar{y}_1, \dots, \bar{y}_m)^T$ stellt den durch das Optimierungsverfahren zu minimierenden Vektor des Anspruchsniveaus dar. Für die Optimierungsaufgabe entsteht eine zusätzliche Restriktion g , welche die Optimierungskriterien f_j berücksichtigt. Ziel ist, die aktuellen Funktionswerte des Vektoroptimierungsproblems möglichst gut an die gegebenen Komponenten des Anspruchsniveaus anzunähern und gleichzeitig das Anspruchsniveau zu senken. Die Formulierungen beruhen auf Überlegungen von SALUKVADZE [65], SATTLER [66] sowie PESCHEL [67].

Die zweite vorgestellte Ersatzzielfunktion \mathbf{Ez}_2 ist eine MinMax-Formulierung der Form

$$\mathbf{Ez}_2[\mathbf{f}(\mathbf{x})] = \max \{ \mathbf{f}(\mathbf{x}) \} = \min! \quad (39)$$

Eine Approximation der im Optimum z.T. sehr un stetigen Funktion ermöglicht die Kreisselmeier-Steinhauser-Funktion nach KREISSELMEIER UND STEINHAUSER [68] oder MLEJNEK [69] in der Form

$$\widetilde{\mathbf{Ez}}_2[\mathbf{f}(\mathbf{x}), \rho] = \max_{j=1}^m [f_j(\mathbf{x})] \approx \frac{1}{\rho} \ln \left(\sum_{j=1}^m e^{(\rho f_j(\mathbf{x}))} \right) = \min! \quad (40)$$

Der numerische Steuerungsfaktor ρ regelt die Güte der Approximation an das zu minimierende Maximierungsproblem aus Gleichung (39).

Die dritte Ersatzzielfunktion \mathbf{Ez}_3 ist durch die gewichtete Summation der Einzelzielfunktionen gemäß

$$\mathbf{Ez}_3 [\mathbf{f}(\mathbf{x})] = \sum_{j=1}^m w_j f_j(\mathbf{x}) = \min! \quad (41)$$

mit $0 \leq w_j \leq 1$ und $\sum_{j=1}^m w_j = 1$ definiert. Die Formulierung basiert auf Arbeiten von PESCHEL [67] und ESCHENAUER in [70].

Die in der vorliegenden Arbeit untersuchten Optimierungskriterien sind

- Minimierungsproblem f_1 : Masse der Struktur
- Minimierungsproblem f_2 : Versagenswahrscheinlichkeit der Struktur gegenüber gewählten Versagenskriterien

Es ergibt sich mit den x_i als Variablen der Tragwerksoptimierung das Vektoroptimierungsproblem

$$\begin{aligned} f_1(\mathbf{x}) &= \min! \\ f_2(\mathbf{x}) &= \min! \end{aligned} \quad (42)$$

Die eingesetzte Ersatzzielfunktion \mathbf{Ez}_3 ist in Abschnitt 7.4.1 definiert. Eine ausführliche Darstellung zu Ersatzproblemen für Vektoroptimierungsaufgaben in der Strukturmechanik findet sich bei SATTLER [66]. Für Fragestellungen der Vektoroptimierung bzw. der Mehrzieloptimierung sei z.B. auf STADLER [70], PESCHEL [67], GÖPFERT UND NEHSE [71] sowie ESCHENAUER ET AL. [72] verwiesen. Hinweise zur Vektoroptimierung mit Evolutionsstrategien finden sich bei KURSAWE [73, 74].

3.3 Mehrgliedrige Evolutionsstrategie

Die Leistungsfähigkeit und Anpassung biologischer Systeme kann nicht aus einer vollständig zufallsgesteuerten Suche globaler Optima heraus erklärt werden. Allein aus der Berücksichtigung früherer Entwicklungsschritte erscheint eine sequentiell ablaufende, auf Erreichtem aufbauende Evolution als Erklärung für die hochentwickelten, individuell ihren Lebensräumen angepaßten Arten und Ökosysteme plausibel. Die aus der natürlichen Evolution abgeleiteten naturanalogen Optimierungsverfahren versuchen, die Funktionsprinzipien ausgewählter Mechanismen der biologischen Evolution für die Optimierung mathematisch-technischer Problemstellungen hinsichtlich einer gegebenen Zielfunktion zu nutzen.

3.3.1 Algorithmus der mehrgliedrigen Evolutionsstrategie

Der Begriff Individuum umfaßt sämtliche Informationen, die zu einem dezidierten Zustand des zu untersuchenden technischen Systems im Parameterraum der Optimierungsaufgabe existieren (Genotyp). Für die Individuen wird teilweise zwischen Objekt- und Strategievariablen unterschieden. Die Koordinatenvektoren der Individuen charakterisieren als Objektvariablen die Lage des Individuums im n-dimensionalen Parameterraum. Parameter, die individuelle Einstellungen der Strategie für ein diskretes Individuum festlegen und damit entscheidenden Einfluß auf den Optimierungsfortschritt haben, werden als Strategievariablen bezeichnet.

Die eingesetzte Variante der mehrgliedrigen Evolutionsstrategie geht für die Generation (z) der Beispielpopulation von μ Elternvektoren aus. Für die erste Generation jeder Population sind die Zustandspunkte der Individuen im Parameterraum durch die vorgegebenen Startwerte festgelegt. Um einen Nachkommen der Generation (z) zu erzeugen, werden $lparen$ Eltern zufällig (gleichverteilt) aus der Elternpopulation bestimmt. Die Auswahl der Eltern findet für jeden Nachkommen individuell statt. Der gewählte Rekombinationsmechanismus kombiniert die Informationen der $lparen$ Eltern zu einem neuen Individuum. Das Individuum wird im Mutationsmechanismus durch Addition einer zufallsbestimmten Zahl stochastisch verändert. Die Abfolge von Evolutionsmechanismen wird für alle λ Nachkommen wiederholt.

Die λ Nachkommen der Generation (z) legen zusammen mit $loldie$ Eltern der Generation (z) die Selektionsmenge fest. Jedem Individuum der Selektionsmenge wird über die zu untersuchende Zielfunktion ein Qualitätswert zugewiesen (Phänotyp). Die beiden Variablenklassen jedes Individuums, Objektvariablen und Strategievariablen, nehmen über ihre Zuordnung zum Individuum indirekt ebenfalls am Selektionsprozeß teil. Die μ besten Individuen der Selektionsmenge bilden die Elternschar der nachfolgenden Generation ($z + 1$). Damit ist die Bearbeitung der Generation (z) der besprochenen Population abgeschlossen. Durch Wiederholung der Vorgehensweise entstehen $maxgen$ Generationen eines Zyklus' der betreffenden Population. Der Algorithmus eines vollständigen Zyklus' wird solange wiederholt, bis ein Abbruchkriterium erfüllt bzw. die vorgegebene, maximale Zyklenzahl $maxzyk$ erreicht ist.

Soll eine Zielfunktion erfolgreich durch Evolutionsstrategien untersucht werden, muß sie zwei Mindestanforderungen gerecht werden (RECHENBERG [75]). Zum einen muß jedem Individuum über die Zielfunktion eindeutig ein Funktionswert zugeordnet werden. Zum anderen hat die Zielfunktion Mindestanforderungen an ihren Ordnungszustand zu genügen. Dieses Glattheitspostulat ist nicht mehr erfüllt, wenn die Funktionswerte zu jeder Variableneinstellung vollkommen zufällig festgelegt wären.

Konvergiert eine Population innerhalb des Optimierungsalgorithmus' gegen einen lokalen Ex-

tremwert, so führt dies unter Umständen zur Stagnation der Population. In Abhängigkeit von der Komplexität der untersuchten Zielfunktion können zusätzliche, übergeordnete Strategiemechanismen wie Populationsparallelität, Migration und Isolation eingesetzt werden. Migration definiert den Austausch von Individuen zwischen ansonsten isolierten Populationen. Über die Parallelität *maxpop* isolierter Populationen, die definiert im Rahmen von Migrationsmechanismen Informationen durch *migras* Migranten austauschen, kann die Variabilität der einzelnen Populationsgenpools erhöht werden. Dies bewirkt eine entscheidende Verbesserung der globalen Konvergenzfähigkeit der mehrgliedrigen Evolutionsstrategie bei der Untersuchung hochdimensional multimodaler Problemstellungen (siehe STRENG [76]). Vorteile der mehrgliedrigen Evolutionsstrategie sind:

- Keine Vorkenntnisse zur Optimierungsaufgabe und keine Gradienteninformationen zur Zielfunktion notwendig.
- Untersuchung multimodaler Problemstellungen möglich.
- Selbstadaptive Anpassung der Strategie an die lokale Topologie der Zielfunktion.

Nachteile der mehrgliedrigen Evolutionsstrategie sind:

- Kein universelles Optimierungsverfahren, d.h. das Verfahren ist nicht in der Lage, ein gesuchtes globales Extremum mit Sicherheit zu lokalisieren.
- Es existiert keine geschlossene Theorie zur mehrgliedrigen Evolutionsstrategie.

Die eingesetzte mehrgliedrige Evolutionsstrategie ist nur aus den Hauptmechanismen Rekombination, Mutation und Selektion aufgebaut.

3.3.2 Programm zur mehrgliedrigen Evolutionsstrategie

Für die in Kapitel 7 beschriebenen Untersuchungen zur Einbindung Neuronaler Netze in Evolutionsstrategien wird eine erweiterte Form des Programms KORR nach SCHWEFEL [77] eingesetzt. KORR simuliert $(\mu \uparrow \lambda)$ -Evolutionsstrategien. Die übergeordneten Strategiemechanismen Populationsparallelität und Migration bleiben unberücksichtigt.

Eine Generation der verwendeten (μ, λ) -Evolutionsstrategie zur Untersuchung der restringierten Optimierungsaufgabe \mathcal{P}_1 in Gleichung (14) hat hinsichtlich der Objektvariablen folgende Form:

1. **Initialisierung:** $z = 0$ und $\mathbf{x} \in \mathbb{R}^n$

$$\text{für alle } k = 1, \dots, \mu: \mathbf{x}_{(E,k)}^{(z)} = \left(x_{(E1,k)}^{(z)}, \dots, x_{(En,k)}^{(z)} \right)^T$$

$$\text{so daß für alle } k = 1, \dots, \mu \text{ gilt: } \begin{cases} g_j \left(\mathbf{x}_{(E,k)}^{(z)} \right) = 0 & \text{für alle } j = 1, \dots, e \\ g_i \left(\mathbf{x}_{(E,k)}^{(z)} \right) \leq 0 & \text{für alle } i = (e+1), \dots, m \end{cases}$$

2. **Diskrete Rekombination aus μ Eltern:**

$$\text{für alle } l = 1, \dots, \lambda \text{ mit } \mathbf{x}_{(N,l)}^{(z)} = \left(x_{(N1,l)}^{(z)}, \dots, x_{(Nn,l)}^{(z)} \right)^T:$$

$$\left. \begin{array}{l} r_i = G [1, \mu] \\ x_{(Ni,l)}^{(z)} = x_{(Ei,r_i)}^{(z)} \end{array} \right\} \text{ für alle } i = 1, \dots, n$$

$G [1, \mu]$ gleichverteilte Auswahl eines Individuums aus Intervall $[1, \mu] \subset \mathbb{N}$

3. **Mutation:**

$$\text{für alle } l = 1, \dots, \lambda: \mathbf{x}_{(Nm,l)}^{(z)} = \mathbf{x}_{(N,l)}^{(z)} + \mathbf{z}_1^{(z)}$$

$$\mathbf{z}_1^{(z)} = \left(z_{(1,l)}^{(z)}, \dots, z_{(n,l)}^{(z)} \right)^T$$

$$\text{für alle } i = 1, \dots, n: z_{(i,l)}^{(z)} = N \left(0, \left(\sigma_{(Ni,l)}^{(z)} \right)^2 \right)$$

4. **Restriktionskontrolle:**

$$\text{für alle } l = 1, \dots, \lambda \text{ muß gelten: } \begin{cases} g_j \left(\mathbf{x}_{(Nm,l)}^{(z)} \right) = 0 & \text{für alle } j = 1, \dots, e \\ g_i \left(\mathbf{x}_{(Nm,l)}^{(z)} \right) \leq 0 & \text{für alle } i = (e+1), \dots, m \end{cases}$$

sonst entsprechend viele Nachkommen l ab Schritt 2 neu erzeugen;

5. **Selektion:**

$$f \left(\mathbf{x}_{(S,k)}^{(z)} \right) \leq f \left(\mathbf{x}_{(S,p)}^{(z)} \right) \begin{cases} \text{für alle } k \in \{1, \dots, i\} \wedge \\ \text{für alle } p \in \{(i+1), \dots, s\} \end{cases}$$

$$\text{mit } s = \lambda + \text{loldie} \quad \mathbf{x}_{(E,k)}^{(z)} = \mathbf{x}_{(S,k)}^{(z)} \text{ für alle } k \in \{1, \dots, \mu\}$$

$z := z + 1$; zurück zu Schritt 2

Hierbei ist $\mathbf{x}_{(E,k)}^{(z)}$ der Vektor des k-ten Elternindividuums der Generation (z), $\mathbf{x}_{(N,l)}^{(z)}$ der Vektor des l-ten Nachkommen nach erfolgter Rekombination, $\mathbf{x}_{(Nm,l)}^{(z)}$ der Vektor des l-ten Nachkommen nach erfolgter Mutation sowie $N \left(0, \left(\sigma_{(Ni,l)}^{(g)} \right)^2 \right)$ eine Normalverteilung mit Mittelwert null und mit der Standardabweichung $\sigma_{(Ni,l)}^{(g)}$ als Schrittweite der Komponente i des l-ten Nachkommenvektors. Zur Realisation innerhalb eines Computerprogramms sind die Restriktionsbedingungen der Punkte 1. und 4. des Algorithmus durch geeignete Toleranzbereiche zu ersetzen.

Für eine detaillierte mathematische Darstellung der mehrgliedrigen Evolutionsstrategie auch hinsichtlich der Evolution der Strategievariablen bzw. der Schrittweitensteuerung sei z.B. auf SCHWEFEL [77] und BÄCK [78] verwiesen. Eigene Erweiterungen des Programms KORR umfassen die gleichzeitige Berücksichtigung mehrerer Optimierungskriterien im Rahmen des multimodalen Vektoroptimierungsproblems durch eine Ersatzzielfunktion sowie die Einbindung trainierter Neuronaler Netze zur Berücksichtigung der Optimierungskriterien (siehe Kapitel 7).

3.4 Gegenüberstellung deterministischer und stochastischer Verfahren

Ein direkter Vergleich zwischen deterministischen und stochastischen Verfahren kann nicht allgemein, sondern immer nur anhand einzelner Problemklassen durchgeführt werden. Beide Klassen von Optimierungsverfahren arbeiten mit konzeptionell unterschiedlichen Algorithmen, was sich direkt auf die Anforderungen der Verfahren an die zu untersuchende Zielfunktion mit zugehörigen Randbedingungen auswirkt.

Grundsätzlich spricht man bei deterministischen Optimierungsverfahren von den Klassen der *Optimality Criteria Methods* (OC) und dem *Mathematical Programming* (MP). Beide Verfahrensklassen sind zueinander korreliert. Bei gleicher Aufgabenstellung und gleichen Approximationen der Problemstellung liefern sie identische Resultate. Die (OC)-Methoden nutzen mathematische oder physikalische Parameter der gesuchten Lösung als Optimierungskriterien eines Lösungsalgorithmus. Dies können z.B. Obergrenzen für Spannungen sein. Es ist dabei aber immer nur die Berücksichtigung einer bestimmten Klasse von Randbedingungen möglich. Die Methoden der mathematischen Optimierung (MP) berücksichtigen dagegen gleichzeitig alle Randbedingungen, unabhängig von ihrer physikalischen Bedeutung. Die Optimierungskriterien sind durch die Kuhn-Tucker-Bedingungen gegeben, während die primäre Optimierungsaufgabe durch rekursive Approximationskonzepte genähert wird. Auch macht man sich spezielle mathematische Eigenschaften der Näherungen wie Separierbarkeit oder Konvexität zu Nutze.

Die grundlegende Strategie der im folgenden erwähnten deterministischen Verfahren beruht auf linearer (konvexer) oder hybrider (linear und invers gemischt) Approximation der im allgemeinen nichtlinearen Problemstellung. Weiter ist in diesem Zusammenhang noch die *Method of Moving Asymptotes* (MMA) zu erwähnen. Die explizite Form der Näherung ermöglicht je nach Approximationsmethode die Auswahl problemangepaßter und damit effektiver Lösungsverfahren. Über eine iterative Abfolge lokaler Approximationen der Zielfunktion sowie der Randbedingungen wird die Konvergenz des Verfahrens gegen die gesuchte Optimallösung

angestrebt. Dieser Algorithmus ist notwendig, da die Strukturantwort als Reaktion auf Belastungssituationen für realistische Strukturen im allgemeinen nicht in einer analytischen, expliziten Form angegeben werden kann, sondern implizit vorliegt und jeweils über eine FEM-Analyse ermittelt werden muß. Dies motiviert den Einsatz Neuronaler Netze. Eine beschränkte Anzahl an FEM-Analysen erlaubt die Generierung von Trainingsdaten zur hinreichend exakten Klassifikation und Erfassung des mechanischen Modells durch problemspezifische oder konventionelle Neuronale Netze. Alternativ ist der Einsatz deterministischer Approximationsmodelle zur Berücksichtigung des zu untersuchenden Strukturmodells denkbar.

Die Methode des *Sequential Quadratic Programming* (SQP) beruht u.a. auf den Arbeiten von HAN [79], POWELL [80] und SCHITTKOWSKI [81].

Die durch FLEURY [82] entwickelte und später durch SVANBERG [83] erweiterte Methode des *Sequential Convex Programming* (SCP) ist ein speziell an die Anforderungen der Tragwerksoptimierung angepaßtes Verfahren.

SVANBERG [83] untersucht Problemstellungen, bei denen das konvexe Optimierungsverfahren der SCP-Methode ungenaue Approximationen der Optimierungsaufgabe liefert. Um den Grad der Konvexität der Approximation individuell an die Problemstruktur anpassen zu können, begrenzt SVANBERG [84] das Unterproblem durch künstliche Randbedingungen (*moving asymptotes*). Die daraus entwickelte *Method of Moving Asymptotes* (MMA) ist eine Verallgemeinerung der individuell direkten bzw. konvexen Approximation, wie sie in der SCP-Methode verwendet wird.

Deterministische Optimierungsmethoden wie z.B. die SQP-Methode hängen in sehr starkem Maße von exakten Gradienteninformationen ab. Rundungsfehler oder notwendige Approximationen höherer Ableitungen haben signifikante Auswirkungen auf Konvergenzverhalten und Effizienz der Verfahren. Die SQP-Methode ist ein allgemeines, deterministisches Optimierungsverfahren. Die SQP-Methode ist bei genauen Gradienteninformationen durch hohe Konvergenzgeschwindigkeit und Genauigkeit in Optimumnähe charakterisiert. Dies ist für realitätsnahe Anwendungen aber meist uninteressant. Die SCP-Methode und die daraus abgeleitete MMA-Methode sind in ihren Approximationsalgorithmen speziell auf die Anforderungen der Tragwerksoptimierung ausgerichtet. Die SCP-Methode liefert auch weit entfernt vom Optimum schnelle Konvergenz gegen das Optimum.

Aus der Klasse der stochastischen Optimierungsverfahren wird in der vorliegenden Arbeit die auf naturanalogen Mechanismen der biologischen Evolution aufbauende, mehrgliedrige Evolutionsstrategie eingesetzt. Dieses und verwandte Verfahren benötigen als einzige Information aus der Optimierungsaufgabe ein Qualitätsmaß zur Klassifizierung der aktuellen Ortsvektoren.

ren ihrer Individuen. Es werden keine Gradienteninformationen der Zielfunktion benötigt. Der sequentielle Charakter der mehrgliedrigen Evolutionsstrategie berücksichtigt aus der Abfolge der Generationen eine Historie der untersuchten Problemstellung. Dies unterscheidet das Verfahren von der reinen Stichprobensuche wie z.B. bei Monte-Carlo-Methoden.

Die Berücksichtigung der Randbedingungen kann bei den untersuchten stochastischen und deterministischen Verfahrensklassen implizit mittels Straffunktionen oder über Multiplikatoransätze mit Hilfe einer erweiterten Lagrange-Funktion erfolgen.

Dort wo spezialisierte, deterministische Verfahren zur Lösung bestimmter Problemklassen existieren, sind diese bezüglich Konvergenzgeschwindigkeit und Genauigkeit der erzielten Lösung stochastischen Verfahren überlegen. Stochastische Verfahren benötigen aus ihrer Konzeption heraus stets mehr Rechenaufwand als Verfahren, die weitergehende Informationen über die Problemstellung nutzen. Deterministische Verfahren sind dagegen unflexibler, stark in den behandelbaren Aufgabenklassen beschränkt und Störungen gegenüber anfälliger. Stochastische Verfahren haben dort ihre Berechtigung, wo keine spezialisierten Algorithmen existieren, wo deterministische Verfahren nicht anwendbar sind (z.B. fehlende Gradienteninformation, stark multimodale Zielfunktionen) oder wo vorab keine Informationen über die zu untersuchende Problemstellung vorliegen.

Es ist unmöglich, eine allgemein gültige Methode zur Bestimmung der globalen Lösung einer nichtlinearen, restringierten und vielfach auch multimodalen Optimierungsaufgabe anzugeben. RINNOOY KAN UND TIMMER [85] stellen fest, daß die Lösung des unbeschränkten globalen Optimierungsproblems nicht mit einer begrenzten Anzahl von Rechenschritten gefunden werden kann. Es können deshalb zwei Schlüsse gezogen werden:

- Die für spezielle Anwendungen in der Tragwerksoptimierung entwickelten deterministischen Verfahren sind den stochastischen Verfahren in diesen Anwendungsfällen überlegen.
- Für die Untersuchung multimodaler Funktionszusammenhänge empfehlen sich stochastische Verfahren. Dies kann aus dem multimodalen Charakter der mit Neuronalen Netzen beschreibbaren Zusammenhänge zwischen Ein- und Ausgabeparametern begründet werden.

Bei der Untersuchung von Optimierungsaufgaben, die mehrere multimodale Optimierungsziele verfolgen, kann das gesuchte globale Extremum durch stochastische Optimierungsverfahren wie der mehrgliedrigen Evolutionsstrategie gefunden werden. Eine Sicherheit hierfür gibt es nicht. Für deterministische Optimierungsverfahren gilt dies nur für Startpunkte aus

einer geeigneten Umgebung um das globale Extremum.

Für die Zukunft wäre ein hybrider Algorithmus aus stochastischen und deterministischen Optimierungsverfahren denkbar. Ein stochastisches Verfahren wie eine mehrgliedrige Evolutionsstrategie könnte die Umgebung um ein Extremum lokalisieren und das dort schnell konvergierende SQP-Verfahren das Extremum dann präzise bestimmen.

Das nächste Kapitel beschreibt die Grundlagen der Zuverlässigkeitstheorie. Damit ermittelte Aussagen zum Versagensverhalten von Tragwerken bilden ein Kriterium der untersuchten Optimierungsaufgabe.

4 Zuverlässigkeitstheorie

Aufgabe der Zuverlässigkeitstheorie ist die Bestimmung einer Kenngröße, mit der die Überlebenswahrscheinlichkeit eines technischen Systems quantifiziert werden kann. Dies gilt in besonderem Maße für Systeme mit hohem Schadenspotential wie z.B. Flugzeuge.

Berücksichtigt man neben deterministischen auch unsichere Systemeigenschaften für die untersuchte Aufgabenstellung, so erfordert eine theoretische Behandlung der Unsicherheiten die Verwendung statistischer sowie wahrscheinlichkeitstheoretischer Methoden. Die Statistik ermöglicht durch die mathematische Aufarbeitung von Beobachtungen der unsicheren Parameter den Einsatz wahrscheinlichkeitstheoretischer Methoden. Modellbildungsprozesse führen zu mechanischen und stochastischen Modellen des untersuchten Systems. Die daraus bestimmbare Versagenswahrscheinlichkeit quantifiziert für unsichere Einwirkungen auf das System bzw. unsichere Systemeigenschaften die Wahrscheinlichkeit, daß einer der für ein Versagen als maßgeblich definierten Grenzzustände überschritten wird. Damit ist auch die Zuverlässigkeit des Systems bekannt.

Die eingesetzte Zuverlässigkeitstheorie gründet sich im wesentlichen auf Ergebnisse des Sonderforschungsbereichs 96 „Zuverlässigkeit der Bauwerke“ (siehe u.a. GRUNDMANN [86] und ABDO SARRAS [87]). In diesen Arbeiten wurden numerische Methoden zur effizienten Abschätzung mehrdimensionaler Wahrscheinlichkeitsintegrale entwickelt, wie sie bei der Berechnung der Versagenswahrscheinlichkeit von Struktursystemen auftreten. Im weiteren wird ein Näherungsverfahren erster Ordnung FORM (**F**irst **O**rdner **R**eliability **M**ethod) mit zugehörigen Algorithmen sowie deren Kopplung an ein bestehendes FEM-Programm eingesetzt. Ein Permutationsalgorithmus im so entstandenen Programm FEMZA (siehe Abschnitt 4.5) steuert die Generierung der zum Training Neuronaler Netze notwendigen Datensysteme.

4.1 Grundlagen der Wahrscheinlichkeitsrechnung

Eine Größe X wird als Zufallsvariable bezeichnet, wenn es sich bei ihrem Wert um eine Größe handelt, die durch einen Auswahlmechanismus zufällig bestimmt ist und Wahrscheinlichkeiten für das Eintreten der mit diesem Ausdruck beschriebenen Ereignisse angebar sind. Bei der Durchführung von Zufallsexperimenten nimmt die Zufallsvariable X den Wert x an, der als Realisation von X bezeichnet wird und dem P als die Wahrscheinlichkeit für die Realisation des betrachteten Ereignisses zugeordnet wird. P ist eine Zahl zwischen null und eins. Die Menge aller möglichen Ergebnisse eines Zufallsexperiments spannt den Ereignisraum auf. Zufallsvariablen können diskret oder kontinuierlich verteilt sein. Zu ihrer Beschreibung ver-

wendet man Wahrscheinlichkeitsdichtefunktionen f_X . Bildet man $f_X(x) \cdot dx$, so ergibt sich die Wahrscheinlichkeit, daß die Zufallsvariable einen Wert in einem x einschließenden Intervall dx annimmt. Im folgenden gelte eine Beschränkung auf stetige Verteilungen. Die Verteilungsfunktion der Zufallsvariablen X ist gegeben durch

$$F_X(t) = P(X \leq t) \quad (43)$$

und nimmt mit Hilfe der nichtnegativen Dichtefunktion f_X die Form

$$F_X(t) = \int_{-\infty}^t f_X(\xi) d\xi \quad \text{bzw.} \quad F_X(t) = \int_0^t f_X(\xi) d\xi \quad (44)$$

an. Die Integrationsgrenze null ist für technische Systeme u.U. physikalisch sinnvoll. Für den Grenzwert $\lim_{t \rightarrow \infty} F_X(t) = 1$ tritt ein Ereignis mit Sicherheit (Wahrscheinlichkeit 1) ein. Wahrscheinlichkeitsverteilungen sind durch Lageparameter mit zugehöriger Dichtefunktion beschrieben.

Eine zur Normalverteilung (Gleichung (45)) gehörende verkürzte Schreibweise mit Erwartungswert μ und Varianz $\text{Var}[X] = \sigma^2$ lautet $N(\mu, \sigma^2)$. Anschaulich bedeutet dies eine Schwankung der Zufallsgröße um den Erwartungswert μ proportional zur Standardabweichung σ . Der zentrale Grenzwertsatz besagt, daß die Verteilungsfunktion der standardisierten Zufallsvariablen Z_n für eine wachsende Zahl n der Zufallsvariablen gegen die Standardnormalverteilung Φ konvergiert (HARTUNG ET AL. [88]). Aus der Dichtefunktion normalverteilter Zufallsvariablen X

$$f_X(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2} \quad (45)$$

folgt durch die lineare Transformation $Y = \frac{X-\mu}{\sigma}$ die Dichtefunktion der Standardnormalverteilung $N(0,1)$

$$\varphi(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} . \quad (46)$$

Die Verteilungsfunktion der Standardnormalverteilung

$$\Phi(y) = \int_{-\infty}^y \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} d\xi \quad (47)$$

liegt damit ebenfalls fest. Ist $\ln(X)$ eine $N(\mu, \sigma^2)$ -verteilte Zufallsvariable, so wird X als logarithmisch normalverteilt bezeichnet. Die Dichtefunktion der Lognormalverteilung folgt zu

$$f_{X,\log}(x) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} & \text{für } x > 0 \\ 0 & \text{sonst .} \end{cases} \quad (48)$$

Im Rahmen einer stochastischen Modellbildung repräsentiert die Normalverteilung eine Summe von Zufallsvariablen und die Lognormalverteilung ein Produkt von Zufallsvariablen.

4.2 Aufgaben der Statistik

Mit Rücksicht auf die formelmäßige Handhabbarkeit werden unsichere oder zufällige Systemparameter, Einwirkungen oder Prozesse durch mathematische Beziehungen in Form von Verteilungsfunktionen beschrieben. Aufgabe der Statistik ist es, aus der meist sehr begrenzten Anzahl von Zufallsexperimenten, Versuchsauswertungen oder statistischen Daten zutreffende stochastische Modelle in Form von Verteilungsfunktionen zu bestimmen. Zur stochastischen Modellbildung ist über die rein statistische Information hinaus Wissen über die zugrundeliegenden, die Unsicherheiten verursachenden physikalischen Mechanismen unverzichtbar.

Im weiteren werden für die unsicheren Systemparameter (Basisvariablen) die Verteilungsfunktionen, d.h. die zugehörigen Wahrscheinlichkeitsdichtefunktionen sowie deren Lageparameter, als bekannt vorausgesetzt.

4.3 Zuverlässigkeitstheorie

Das im folgenden beschriebene Verfahren ermittelt eine operative Versagenswahrscheinlichkeit P_f des untersuchten Systems. Diese ist nicht mit der relativen Häufigkeit von Versagensereignissen in der Natur identisch. P_f wird häufig als Fraktile der Standardnormalverteilung angegeben. Man spricht dann von dem Zuverlässigkeitsindex

$$\beta = -\Phi^{-1}(P_f) . \quad (49)$$

Basisvariablen sind unsichere, elementare Grundgrößen des Systems, die durch stochastische Modelle in ihrem Verhalten beschrieben werden. Als Beispiele seien unsichere Materialkenngrößen, Geometrieabmessungen oder Lasten genannt.

Stochastische Modelle beschreiben die unsicheren Basisvariablen und ihre Interaktion bei der Ermittlung des Versagensverhaltens des untersuchten Systems. Das mechanische Modell beschreibt die Systemanalyse. Es liefert Systemantworten als Reaktion auf äußere Einwirkungen sowie aktuelle sichere und unsichere Systemeigenschaften. Hierzu wird für die untersuchten Fachwerkstrukturen die Finite-Elemente-Methode verwendet. Die Festlegungen im Rahmen des stochastischen und mechanischen Modellbildungsprozesses bestimmen, welche Grundgrößen die Problemstellung hinreichend beschreiben und welche der Grundgrößen als unsicher oder als deterministisch zu berücksichtigen sind. Die Kopplung beider Modelle ist in Abschnitt 4.5 beschrieben. Eine Lastgeschichte oder zeitliche Änderungen der Zufallsvariablen (z.B. Ermüdung, Kriechen) wird nicht berücksichtigt.

Die Systemantwort des mechanischen Modells

$$\mathbf{mmod}(\mathbf{X}) \quad \text{mit } \mathbf{X} = (X_1, X_2, \dots, X_n)^T \quad (50)$$

als Reaktion auf die unsicheren Basisvariablen \mathbf{X} definiert zusammen mit der Systemantwort des mechanischen Modells der deterministischen Parameter \mathbf{d} des untersuchten Systems (z.B. gesicherte Festigkeitsgrößen oder feste Lasten) den Zustandsvektor $\mathbf{Z} = (\mathbf{mmod}(\mathbf{X}), \mathbf{mmod}(\mathbf{d}))$. Beispiele für Systemantworten des mechanischen Modells sind Stabspannungen und Knotenverschiebungen.

P_f ist abhängig von Annahmen zu den Modellen für die stochastische Beschreibung der Basisvariablen des untersuchten Systems (stochastisches Modell), von Annahmen zum mechanischen Modell des Systems und der Festlegung der in den Grenzzustandsfunktionen berücksichtigten Versagensereignisse.

Die Zuverlässigkeit P_s ist die Wahrscheinlichkeit dafür, daß keiner der für ein Versagen maßgeblichen Grenzzustände erreicht bzw. überschritten wird. Komplementär hierzu ist die Versagenswahrscheinlichkeit P_f . Sie quantifiziert die Wahrscheinlichkeit, daß das System als Reaktion auf die unsicheren Eigenschaften der Basisvariablen und die Systembelastung einen oder mehrere der für Versagen maßgeblichen Grenzzustände erreicht bzw. überschreitet. Damit ergibt sich

$$P_f = 1 - P_s \quad (51)$$

Das Versagen eines technischen Systems, z.B. eines Stabfachwerks ist systembedingt durch verschiedene Kombinationen von Ausfällen seiner Komponenten definiert. Bei statisch bestimmten Fachwerken führt bereits das Versagen einer Komponente (Stab) zu Systemversagen. Im Zusammenhang mit der Zuverlässigkeitsuntersuchung von statisch bestimmten Fachwerkstrukturen kann damit von Seriensystemen gesprochen werden. Für diese Seriensysteme (ODER-Verknüpfung) ist Systemversagen durch die Vereinigung der Versagensbereiche der Einzelkomponenten V_i zu

$$V_{f \text{ Serie}} = \bigcup_{i=1}^p V_i \quad (52)$$

gegeben. Bei statisch unbestimmten und damit redundanten Fachwerken müssen erst mehrere Komponenten jeweils nach Lastumlagerung ausfallen, um Systemversagen zu verursachen.

Die allgemeine logische Definition des Systemversagensbereichs V kann für zeitinvariante Probleme durch die Modellbildung als Seriensystem mit parallelen Untersystemen

$$V = \bigcup_{i=1}^s \bigcap_{j=1}^{p_i} V_{ij} \quad (53)$$

erfolgen. Die Variable s beschreibt die Anzahl der in Serie angeordneten parallelen Untersysteme, p_i die Anzahl der Komponenten im parallelen Untersystem und V_{ij} den Versagensbereich der Komponente j im Untersystem i (siehe ABDO SARRAS [87] sowie HOHENBICHLER UND RACKWITZ [89]). Die zugehörige Systemversagenswahrscheinlichkeit folgt zu

$$P_f = P(V) . \quad (54)$$

Im weiteren sei die Systemversagensfunktion, d.h. die gemeinsame, stetige Verteilungsfunktion aller Basisvariablen, als bekannt vorausgesetzt. Es wird weiter vorausgesetzt, daß die Verteilungsfunktion von \mathbf{X} bis auf endlich viele Sprungstellen stetig und differenzierbar ist. Damit existiert die zugehörige Verteilungsdichte $f_{\mathbf{X}}(\mathbf{X})$. Dies ist eine Voraussetzung für die Anwendbarkeit der FORM-Methode (siehe Abschnitt 4.4).

Das Versagen einzelner Komponenten des untersuchten Systems ist durch das Versagenskriterium g als Funktion des Zustandsvektors \mathbf{Z} definiert. Die Zustandsfunktion $g(\mathbf{Z})$ teilt den durch \mathbf{Z} aufgespannten Ereignisraum $\mathbb{E}_{\mathbf{Z}}$ der Zustandsvariablen in einen Versagens- und einen Überlebensbereich. Der Versagensbereich ist durch $V = \{\mathbf{Z} : \text{für alle } g(\mathbf{Z}) \leq 0\}$ und der Überlebensbereich (sichere Bereich) durch $S = \{\mathbf{Z} : \text{für alle } g(\mathbf{Z}) > 0\}$ definiert. Die Grenzzustandsfunktion ist durch $g(\mathbf{Z}) = 0$ gegeben. Im weiteren wird angenommen, daß $g(\mathbf{Z}) = 0$ lokal stetig und zweifach differenzierbar sei. Die Zustandsfunktion $g(\mathbf{Z})$ definiert den aktuell betrachteten Versagensfall und ist eine explizite Funktion des Vektors \mathbf{Z} , der über das mechanische Modell indirekt als Funktion des Vektors \mathbf{X} angesehen werden kann.

Die speziellen statistischen Eigenschaften der unsicheren Basisvariablen sind über die Dichte der Systemversagensfunktion $f_{\mathbf{X}}(\mathbf{X})$ berücksichtigt. Die zugehörige Versagenswahrscheinlichkeit ist durch

$$P_f = P(V) = P(g(\mathbf{Z}) \leq 0) = \int_V f_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} = \int_{g(\mathbf{Z}) \leq 0} f_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad (55)$$

gegeben. Die so ermittelte Versagenswahrscheinlichkeit ist immer in bezug auf die zugrundeliegenden Annahmen und die aktuell berücksichtigten Versagensfälle der Zustandsfunktion $g(\mathbf{Z})$ zu sehen. Sind deren physikalische Grundlagen unklar oder treten nicht berücksichtigte Versagensfälle auf, so sind von der Zuverlässigkeitstheorie keine gesicherten Aussagen zu erwarten.

Die gemeinsame Wahrscheinlichkeitsfunktion $P = P(V) + P(S)$ beschreibt über dem durch die Zustandsvariablen \mathbf{Z} aufgespannten n -dimensionalen Ereignisraum $\mathbb{E}_{\mathbf{Z}} \subseteq \mathbb{R}^n$ einen n -dimensionalen Graph in \mathbb{R}^{n+1} über \mathbb{R}^n mit zugehörigem Volumen 1. Das Volumen wird durch die Grenzzustandsfunktion in den Überlebensbereich S und den Versagensbereich V

aufgeteilt. Der Volumenanteil von V quantifiziert die Versagenswahrscheinlichkeit P_f zugehörig zu den im stochastischen und mechanischen Modell getroffenen Annahmen.

Analytische Lösungen der Integrale sind nur für einfachste Verteilungs- und Grenzversagensfunktionen berechenbar. Numerische Lösungen sind in realistischen Fällen wegen der damit verbundenen großen Dimension der Problemstellung und des im allgemeinen komplizierten Verlaufs der Zustandsfunktion g nur mit sehr hohem Rechenaufwand realisierbar. Deshalb soll im folgenden ein Näherungsverfahren erster Ordnung zur Lösung der bei der Berechnung von P_f auftretenden Volumenintegrale beschrieben werden (siehe z.B. Gleichung (55)). Weitere Grundlagen zu dieser Thematik finden sich z.B. bei HARTUNG ET AL. [88] und zur Zuverlässigkeitstheorie insbesondere bei GRUNDMANN [86] und ABDO SARRAS [87].

4.4 FORM

Die FORM-Methode liefert eine Näherung erster Ordnung der Versagenswahrscheinlichkeit aus Gleichung (55) (siehe z.B. COMREL MANUAL [90]).

4.4.1 Methodik

Im folgenden Abschnitt wird die Approximation der Versagenswahrscheinlichkeit P_f für den allgemeinen, n -dimensionalen Fall der Zustandsfunktion $g(\mathbf{Z})$ unter Verwendung der FORM-Methode dargestellt.

FORM arbeitet in dem durch unabhängige, standardnormalverteilte Basisvariablen aufgespannten Ereignisraum \mathbb{E}_U . Basisvariablen, die statistisch durch beliebige andere Verteilungsfunktionen beschrieben sind, werden bzgl. ihrer stochastischen Aussage äquivalent (Wahrscheinlichkeitsinhalt) in den Ereignisraum standardnormalverteilter Basisvariablen \mathbb{E}_U transformiert. Die Lösung einer in \mathbb{E}_U definierten Minimierungsaufgabe bestimmt den Punkt auf der Grenzzustandsfunktion, der dem Koordinatenursprung im Ereignisraum \mathbb{E}_U am nächsten liegt. Die in diesem Punkt durch eine Taylorentwicklung erster Ordnung linearisierte Grenzzustandsfunktion bildet die Integrationsgrenze des Volumenintegrals aus Gleichung (55) und erlaubt somit eine gute Abschätzung der Versagenswahrscheinlichkeit des untersuchten Systems. Die FORM-Methode hat den entscheidenden Vorteil einer nur linearen Abhängigkeit ihres Rechenaufwands von der Dimension des Basisvariablenvektors \mathbf{X} . Die theoretischen Voraussetzungen dieser Methodik beruhen u.a. auf HASOFER UND LIND [91] durch die Definition eines bezüglich der Formulierung der Zustandsfunktion g invarianten Zuverlässigkeitsindizes' β sowie auf RACKWITZ [92] durch eine direkte Darstellung der Basisvariablen über

zugehörige, beliebige Verteilungsfunktionen.

Die gültige Systemversagensfunktion ergibt sich gemäß Gleichung (53) als Kombination der zur aktuellen Problemstellung berücksichtigten Zustandsfunktionen g_{ij} der Komponente j im Untersystem i zu

$$P_f = P \left(\bigcup_{i=1}^s \bigcap_{j=1}^{P_i} [g_{ij} \leq 0] \right). \quad (56)$$

Die vorliegende Arbeit ist bzgl. der Untersuchung des Zuverlässigkeitsverhaltens von Fachwerkstrukturen auf statisch bestimmte Systeme beschränkt. Für Untersuchungen zum Versagensverhalten handelt es sich damit um Seriensysteme. Für diese vereinfacht sich Gleichung (56) zu

$$P_{f \text{ Serie}} = P \left(\bigcup_{i=1}^s [g_i \leq 0] \right). \quad (57)$$

Eine Abschätzung der Versagenswahrscheinlichkeit eines Seriensystems $P_{f \text{ Serie}}$ kann mit $P_i = P\{g_i \leq 0\}$ durch

$$\max \{P_i\} \leq P_{f \text{ Serie}} \leq \sum_{i=1}^s P_i \quad (58)$$

gegeben werden. Die durch FORM bestimmte Versagenswahrscheinlichkeit P_f ist ausschließlich durch die Definition des Systemversagens gemäß Gleichung (56) bzw. (57) sowie durch die Festlegung der einzelnen Zustandsfunktionen g_{ij} bzw. g_i bestimmt.

4.4.2 Rosenblatt-Transformation

Das zur Approximation der Versagenswahrscheinlichkeit P_f eingesetzte Näherungsverfahren FORM arbeitet im Ereignisraum unabhängiger, standardnormalverteilter Basisvariablen \mathbb{E}_U . Unsichere Basisvariablen X_j , die durch beliebige andere Verteilungsfunktionen beschrieben sind und in allgemeiner Abhängigkeit zueinander stehen (bedingte Wahrscheinlichkeiten), werden bzgl. ihrer Wahrscheinlichkeitsaussage äquivalent in unabhängige, standardnormalverteilte Basisvariablen U_j transformiert.

Die Grundlage hierzu bilden Untersuchungen von HOHENBICHLER UND RACKWITZ [89]. Sie verwenden die bzgl. der Wahrscheinlichkeitsaussage invariante (punktweise Wahrscheinlichkeitstreue) Rosenblatt-Transformation T_R (ROSENBLATT [93]) allgemeiner Basisvariablen X_j (beliebige Verteilungsfunktion) in voneinander unabhängige, standardnormalverteilte Variablen U_j . Die Vorschrift der inversen Rosenblatt-Transformation T_R^{-1} ist durch

$$\mathbf{u} = T_R^{-1}(\mathbf{x}) = \left(T_1^{-1}(x_1), T_2^{-1}(x_1, x_2), \dots, T_n^{-1}(x_1, \dots, x_n) \right)^T \quad (59)$$

gegeben. Im Raum \mathbb{E}_U der unabhängigen, standardnormalverteilten Basisvariablen kann dann die FORM-Methode zur Approximation der Systemversagenswahrscheinlichkeit eingesetzt werden.

4.4.3 Approximation der Versagenswahrscheinlichkeit

Die Wahrscheinlichkeit, daß ein Ereignis A im Ereignisraum \mathbb{E}_U eintritt, ist durch dessen Ursprungsentfernung $\beta = \|\mathbf{U}_A\|$ bestimmt. Mit Gleichung (49) folgt über die FORM-Methode

$$P_f = \Phi(-\beta) . \quad (60)$$

Der dem Koordinatenursprung nächstgelegene Punkt \mathbf{U}^* auf dem Graph der Grenzzustandsfunktion

$$\mathbf{U}^* = \tilde{\mathbf{g}}(\mathbf{U}) = \mathbf{g}(\mathbf{T}_R^{-1}(\mathbf{U})) = \mathbf{g}(\mathbf{Z}) = 0 \quad (61)$$

in \mathbb{E}_U wird als Entwicklungspunkt E bezeichnet. Voraussetzung ist $\tilde{\mathbf{g}}(\mathbf{0}) > 0$, d.h. der Ursprung liegt im sicheren Bereich S . Für die Bestimmung des Entwicklungspunkts in nichtlinearen Fällen von $\tilde{\mathbf{g}}(\mathbf{U})$ bzgl. \mathbf{U} wird ein iteratives Gradientenverfahren nach RACKWITZ UND FIESSLER [94] sowie SCHITTKOWSKI [81] eingesetzt. Der Algorithmus bestimmt den Entwicklungspunkt E auf der Grenzzustandsfunktion $\tilde{\mathbf{g}}(\mathbf{U}) = 0$. Die Zustandsfunktion $\tilde{\mathbf{g}}$ bestimmt über die Randbedingung

$$\min \{ \|\mathbf{U}\| \text{ für alle } \tilde{\mathbf{g}}(\mathbf{U}) \leq 0 \} \quad (62)$$

das durch die FORM-Methode zu berechnende Volumen des Versagensbereichs V unter der Wahrscheinlichkeitsdichtefunktion des Gesamtsystems in Gleichung (55). Im Ereignisraum \mathbb{E}_U der unabhängigen, standardnormalverteilten Variablen U_j wird die Grenzzustandsfunktion $\tilde{\mathbf{g}}(\mathbf{U}) = 0$ im Entwicklungspunkt E durch FORM in einer Taylorentwicklung erster Ordnung linear approximiert. Dabei handelt es sich um eine n -dimensionale Hyperebene L in \mathbb{R}^{n+1} über $\mathbb{E}_U \subset \mathbb{R}^n$, die tangential in E an $\tilde{\mathbf{g}}$ anliegt. Sie ist durch

$$L(\mathbf{U}) = \alpha^T \mathbf{U} + \beta \quad (63)$$

über den durch

$$\alpha = - \frac{\mathbf{U}_E}{\|\mathbf{U}_E\|} \quad (64)$$

im Punkt E in \mathbb{E}_U gegebenen normalisierten Vektor α definiert. Damit ist α der Einheitsvektor von E zum Ursprung des Koordinatensystems. Der euklidische Abstand zwischen E und dem Koordinatenursprung entspricht dem Zuverlässigkeitsindex

$$\beta = \|\mathbf{U}_E\| = \min \{ \|\mathbf{U}\| \text{ für alle } \tilde{\mathbf{g}}(\mathbf{U}) \leq 0 \} . \quad (65)$$

Die Berechnung des Volumenintegrals aus Gleichung (55) wird damit über die FORM-Methode durch die Optimierungsaufgabe in Gleichung (65) ersetzt. Der Zuverlässigkeitsindex β ergibt sich aus der Minimierung von $\|\mathbf{U}\|$ unter Berücksichtigung der Randbedingung $\tilde{g}(\mathbf{U}) \leq 0$. Mit Gleichung (63) folgt aus der Linearisierung der Grenzzustandsfunktion $\tilde{g}(\mathbf{U}) = 0$ im Entwicklungspunkt \mathbf{E} der gesuchte Ortsvektor des Entwicklungspunkts zu $\mathbf{U}_E = -\beta\boldsymbol{\alpha}$. Die Versagenswahrscheinlichkeit des betrachteten Systems kann damit unter Berücksichtigung der Länge des normalisierten Vektors $\|\boldsymbol{\alpha}\| = 1$ durch

$$P_f \cong P(\boldsymbol{\alpha}^T \mathbf{U} + \beta \leq 0) = \Phi(-\beta) \quad (66)$$

approximiert werden.

Der euklidische Abstand zwischen \mathbf{U}_E und dem Koordinatenursprung im Ereignisraum \mathbb{E}_U ist nach Lösung der Optimierungsaufgabe in Gleichung (65) minimal und die zugehörige Versagenswahrscheinlichkeit aus der Dichtefunktion der Standardnormalverteilung damit maximal. Grund ist die Rotationssymmetrie der Standardnormalverteilung zum Koordinatenursprung. Der wesentlichste Beitrag zur Versagenswahrscheinlichkeit ist somit von dem Punkt auf der Grenzzustandsfunktion zu erwarten, der dem Ursprung am nächsten liegt. Dies ist der Entwicklungspunkt \mathbf{E} . Gleichzeitig nehmen die Werte der Standardnormalverteilung mit zunehmendem Abstand vom Ursprung proportional zu $e^{-\frac{1}{2}\|\mathbf{U}\|^2}$ ab. Die Methodik garantiert eine konservative Abschätzung der Versagenswahrscheinlichkeit des untersuchten Systems, da kein Punkt auf $\tilde{g}(\mathbf{U}) = 0$ existiert, der einen geringeren Abstand zum Ursprung des Koordinatensystems und damit eine größere Versagenswahrscheinlichkeit besitzt. Der Fehler dieser Näherung ist nicht bestimmbar.

Weiterführende Literatur zur Zuverlässigkeitstheorie findet sich z.B. bei MADSEN ET AL. [95], ADELI [96], SCHUËLLER [97], THOFT-CHRISTENSEN UND BAKER [98] sowie bei SPAETHE [99].

4.5 Kopplung von mechanischem und stochastischem Modell

Der vorliegende Abschnitt beschreibt den Aufbau des FEM-Zuverlässigkeits-Programms FEMZA (Finite-Elemente-Methode und ZuverlässigkeitsAnalyse). Die als Source-Code verfügbaren Programme FEA (siehe Abschnitt 5.3) und FORM (GOLLWITZER ET AL. [100] bzw. COMREL MANUAL [90]) ermöglichen die Entwicklung von FEMZA. FEMZA ist in der Lage, zugehörig zu den Annahmen des stochastischen Modells **smod** und des mechanischen Modells **mmod** Aussagen zur Zuverlässigkeit des untersuchten statischen Systems zu machen. Dies hat folgende Vorteile:

- Permutationen sind zur Erzeugung von Test- und Trainingsdaten realisierbar.
- Parameterstudien sind automatisierbar.

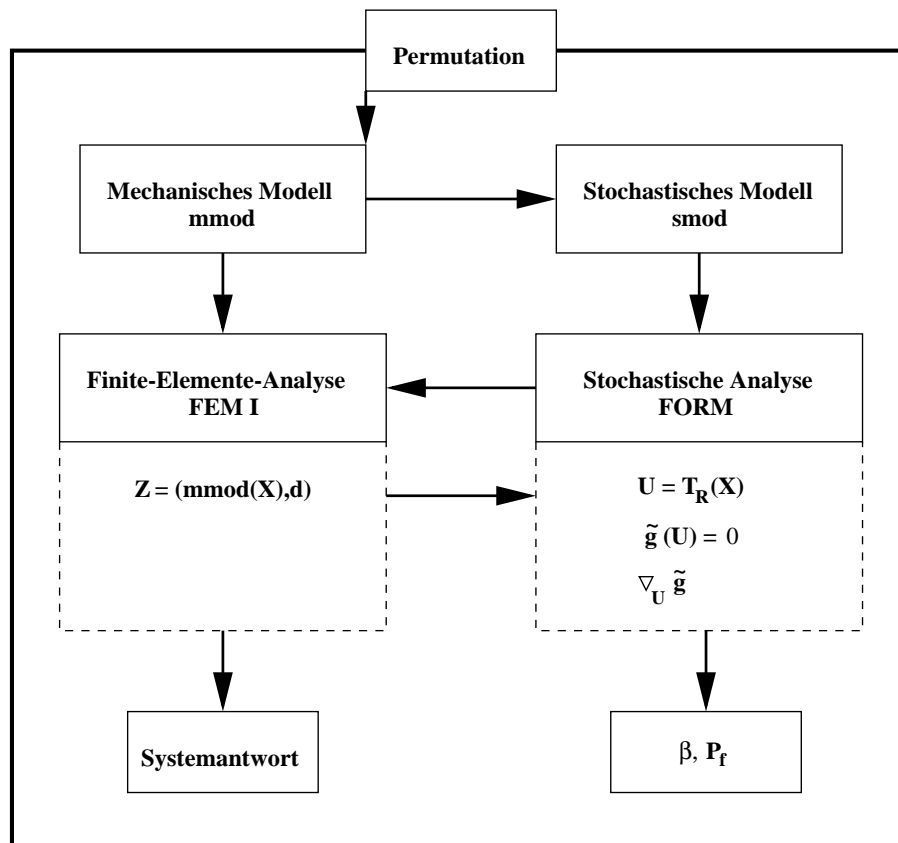


Abbildung 2: Programmablaufplan FEMZA

Das Programm FEMZA (siehe Abbildung 2) gliedert sich im wesentlichen in drei Teile:

- Permutationsalgorithmus
- Mechanisches Modell **mmod** mit FEM-Analyse
- Stochastisches Modell **smod** mit FORM-Analyse

Der Permutationsalgorithmus variiert sukzessive ausgewählte Parameter des mechanischen Modells der untersuchten Problemstellung und erzeugt so die Datensysteme zu Test und Training Neuronaler Netze. Die zu permutierenden Parameter können Materialdaten wie Dichten oder E-Moduli, äußere Lasten oder geometrische Abmessungen wie z.B. Stabdurchmesser sein. Jeder der Parameter wird individuell um eine gegebene Anzahl an Permutationsschritten mit einer zugehörigen Schrittweite inkrementiert. Der Algorithmus berücksichtigt dabei

alle Kombinationsmöglichkeiten der permutierten Parameter untereinander (siehe Abschnitt 5.3.2). Sind permutierte Parameter im stochastischen Modell als unsicher definiert, wird über die Kopplung von mechanischem und stochastischem Modell zu jedem Permutationsschritt eine zugehörige Zuverlässigkeitsaussage ermittelt.

Das stochastische Modell **smod** legt die aktuelle Realisation der unsicheren Basisvariablen des Vektors \mathbf{X} über die Auswahl zugehöriger Wahrscheinlichkeitsdichtefunktionen und deren Lageparameter μ und σ fest. Der Lageparameter μ ist in der vorliegenden Arbeit durch das mechanische Modell über die Kopplung an den Permutationsalgorithmus vorgegeben.

Das mechanische Modell **mmod** definiert das untersuchte statische System für die FEM-Analyse. Der übergeordnete Permutationsalgorithmus steuert Anzahl und Parameter der FEMZA-Berechnungen über die Variation der zu permutierenden Parameter und die Realisation von \mathbf{X} in **smod**. Die Rechnungen liefern zugehörig zur aktuellen Realisation von \mathbf{X} und \mathbf{d} die Systemantwort der Struktur. Damit liegt auch der Zustandsvektor \mathbf{Z} fest.

Nach Bestimmung des Vektors \mathbf{U} mit Hilfe der Rosenblatt-Transformation gemäß Gleichung (59) und des Gradienten der Zustandsfunktion \tilde{g} kann über den Iterationsalgorithmus nach RACKWITZ UND FIESSLER [94] sowie SCHITTKOWSKI [81] der Entwicklungspunkt berechnet werden. Über den euklidischen Abstand $\|\mathbf{U}_E\|$ des Entwicklungspunkts vom Koordinatenursprung liegt der Zuverlässigkeitsindex β und damit die Näherung der Versagenswahrscheinlichkeit P_f fest. Der übergeordnete Permutationsalgorithmus steuert die Variation der zu permutierenden Parameter des mechanischen und damit auch des stochastischen Modells und erzeugt so die Datensysteme zu Test und Training Neuronaler Netze.

Eine stichprobenmäßige Kontrolle der eigenen Ergebnisse erfolgt mit dem kommerziellen Software-Paket COMREL 7(β) der Firma RCP (COMREL MANUAL [90]). Da sowohl FORM5 als auch COMREL 7(β) auf den beschriebenen theoretischen Grundlagen aufbauen, ist COMREL 7(β) sehr gut zur Kontrolle der Ergebnisse aus FEMZA geeignet. Die durchgeführten Testrechnungen zeigen sehr gute Übereinstimmungen der Ergebnisse aus COMREL 7(β) mit denen aus FEMZA.

Mit den geschaffenen Werkzeugen besteht die Möglichkeit, das Versagensverhalten von Tragstrukturen zu beschreiben und dies als ein Kriterium der untersuchten Optimierungsaufgabe zu nutzen. Das folgende Kapitel erläutert Grundlagen zum Themenkomplex Neuronale Netze und neue Überlegungen in den Bereichen problemspezifische Ansatzfunktionen sowie *Pruning* Neuronaler Netze. Ziel ist die Erfassung von Kriterien und Restriktionen der zu untersuchenden Optimierungsaufgaben durch geeignete Neuronale Netze.

5 Neuronale Netze

Neuronale Netze sind massiv parallele, lernfähige informationsverarbeitende Systeme, die in ihren Grundprinzipien nach Aufbau und Systematik den Gehirnen von Säugetieren nachgeahmt sind.

Das menschliche Gedächtnis funktioniert als Assoziativspeicher. Das Erkennen partieller Merkmale eines gespeicherten Objekts löst ein Erinnern an ähnliche oder verwandte Speicherobjekte aus. Von Interesse sind die Aspekte biologischer Neuronenstrukturen (siehe z.B. KANDEL ET AL. [101]), die für Lernen, Speicherung von Daten sowie die Generalisierung und Abstraktion gesammelten Wissens und Erfahrungen verantwortlich sind. Die untersuchten Neuronalen Netze werden dabei nicht aufwendig jeweils als elektronische VLSI-Schaltungen realisiert, sondern durch eine spezielle Software auf konventionellen Computern simuliert.

Im folgenden Kapitel kommen die Simulationssoftware sowie Grundbegriffe und charakteristische Eigenschaften Neuronaler Netze zur Sprache. Weiter werden zwei grundlegend unterschiedliche Philosophien für die Generierung problemangepaßter Neuronaler Netze, die Erzeugung von Test- und Trainingsdaten, die in der vorliegenden Arbeit zum Netztraining verwendeten Lernverfahren sowie das *Pruning* von allgemein angelegten Ausgangsnetztopologien besprochen. Überlegungen zur Komplexität und Leistungsfähigkeit Neuronaler Netze vervollständigen die Ausführungen dieses Kapitels. Für weitergehende Informationen zum Thema Neuronale Netze sei z.B. auf ZELL [23], ROJAS [102] und HASSOUN [103] verwiesen.

Für die in dieser Arbeit durchgeführten Untersuchungen Neuronaler Netze wird die Simulationssoftware *Stuttgart Neural Network Simulator* (SNNS) in der Version 4.1 nach ZELL ET AL. [104] eingesetzt.

5.1 Grundlagen

Der Aufbau von Neuronalen Netzen, wie sie auf Computern simuliert werden, gliedert sich allgemein in eine Eingabeschicht, eine oder mehrere verdeckte Schichten und eine Ausgabeschicht. Grundbausteine der einzelnen Schichten sind sogenannte Prozesselemente (Neuronen), die in ihrem Potential, Informationen zu verarbeiten, durch Zuweisung mathematischer Funktionen veränderlich sind (siehe Abbildung 3 und 4). Die Verbindung der Grundelemente der einzelnen Schichten ist sowohl in ihrer Ausprägung (Gewichtung) als auch nach der Orientierung des Datenflusses variabel. Zusätzlich kann die Vernetzungstopologie der Neuronen

in ihrer Struktur individuell der jeweils zu untersuchenden Problemstellung angepaßt werden.

Durch die Implementierung von Funktionsansätzen in die Netzwerkstruktur ist die Vermittlung von bekanntem Basiswissen über die aktuelle Problemstellung in das Neuronale Netz vor dem Start des eigentlichen Lernverfahrens möglich.

Grundsätzlich ist bei Neuronalen Netzen zwischen Trainings- und Nutzungsphase (Testphase) zu unterscheiden. Der Lernalgorithmus adaptiert in der Trainingsphase die variablen Parameter des Netzes an den inhärent in den Trainingsdaten repräsentierten Zusammenhang. Die Testphase dient alleinig der Einschätzung des Lernerfolgs und der Generalisierungsfähigkeit als Reaktion auf bislang unbekannte Testdaten und beläßt die getestete Netztopologie und Gewichtsmatrix vollkommen unverändert.

Der Lernvorgang verändert die variablen Parameter der Neuronenstruktur als Reaktion auf die an Ein- und Ausgabeschicht anliegenden Daten der Trainingsmuster. Unterschreitet der Fehler zwischen Soll- und Istausgabe (Netzfehler) einen gewählten Grenzwert, so ist der Zusammenhang der Trainingsdaten T mit einer bestimmten Genauigkeit implizit in den Variablen der Neuronalen Netztopologie enthalten. Die Abstraktion der impliziten Übertragungsfunktion aus dem trainierten Netz in eine explizite mathematische Form ermöglicht die Untersuchung der in den Lerndaten repräsentierten Problemstellung mit etablierten mathematischen Methoden.

Im folgenden sollen häufig verwendete Begriffe definiert werden:

Neuron: Auch Prozesselement oder Einheit genannt; kleinste Informationsverarbeitungseinheit in einem Neuronalen Netz.

Transferfunktion: Gesamtfunktionalität eines Neurons; zusammengesetzt aus Eingangs-, Aktivierungs- und Ausgabefunktion (siehe Abbildung 3(links)).

Verbindungen: Auch *links* genannt; gerichtete und mit variabler Gewichtung versehene Informationswege zwischen Neuronen.

Gewichte: Veränderliche, individuelle Wichtung der Verbindungen zwischen Neuronen; werden auch als freie Parameter des Netzes bezeichnet und durch das Netztraining an die Problemstellung angepaßt.

Neuronale Netztopologie: Beschreibt den Aufbau eines Neuronalen Netzes nach Anzahl, Anordnung und Netzplan der Neuronen; die Neuronen werden nach ihrer relativen Position im Neuronalen Netz in ein Schichtenmodell mit Eingabe- und Ausgabeneuronen sowie verdeckten Neuronen eingeteilt (siehe Abbildung 4). Ein Neuronales Netz kann als gerichteter, gewichteter Graph angesehen werden. Die Kanten stellen dabei die gewichteten Verbindungen zwischen den Neuronen dar.

Netzübertragungsfunktion: Durch die Neuronale Netztopologie gegebener mathematischer Zusammenhang zwischen Ein- und Ausgabeneuronen.

Feedforward-Netze: Informationsverarbeitung findet in den gerichteten Verbindungen nur ausgehend von den Eingabeneuronen rückkopplungsfrei in Richtung auf die Ausgabeneuronen statt.

Vollständig vernetzt: *Feedforward*-Netze, bei denen jedes Neuron einer Schicht mit jedem Neuron der bezüglich der Informationsverarbeitung nachfolgenden Schicht (Vorwärtsorientierung) verbunden ist.

Netzplan: Legt bei nicht vollständig vernetzten Netztopologien fest, welche Neuronen mit welchen Neuronen der jeweils nachfolgenden Schicht verbunden sind.

Problemspezifische Neuronale Netze: auch Ähnlichkeitsnetze genannt; Neuronale Netze die bzgl. ihrer Netztopologie so angelegt sind, daß sie eine mathematische Ansatzfunktion exakt beschreiben; möglich sind Regressions- bzw. Interpolationsansätze sowie die Abbildung diskreter Zusammenhänge (siehe Abschnitt 5.2.3 und 5.2.4).

Pruning: Ausdünnungsverfahren zur Reduzierung allgemeiner Ansatztopologien auf minimal notwendige Netztopologien zugehörig zur jeweiligen Problemstellung. Das zugrundeliegende Verb *to prune* kann mit beschneiden oder zurechtstutzen übersetzt werden.

Trainingsdaten: Auch Trainingsmuster genannt; zusammengehörige Vektorpaare, die einen durch ein Neuronales Netz zu erfassenden Zusammenhang f beschreiben und zum Training Neuronaler Netze eingesetzt werden. Als Beispiel seien Vektorpaare $(\mathbf{x}_i, f(\mathbf{x}_i))$ genannt.

Testdaten: Die Muster dienen nach Abschluß der Lernphase der Kontrolle des Lernerfolgs des Netztrainings und werden dem zu testenden Neuronalen Netz nur zu diesem Zweck präsentiert.

5.2 Netztopologien

Generell kann der Aufbau Neuronaler Netze in Mikro-, Meso- und Makrostrukturen eingeteilt werden. Dabei beschreibt die Mikrostruktur die Datenverarbeitung auf der Ebene der Neuronen durch Eingabe-, Aktivierungs- und Ausgabefunktionen. Die Mesostruktur faßt die Mikrostruktur der Neuronen geordnet nach Anzahl der Neuronenschichten, Anzahl der Neuronen in den Schichten, Aufbau der Verbindungstopologie und Art der Verbindungen zusammen. Die Makrostruktur als oberste Stufe der Hierarchie integriert mehrere Neuronale Netze in ein Gesamtsystem, sollte die Komplexität der Aufgabenstellung dies erfordern.

5.2.1 Nomenklatur

In der vorliegenden Arbeit werden ausschließlich *Feedforward*-Netzwerke untersucht. Die Transferfunktion eines Standardneurons f_{trans} ergibt sich aus der sequentiellen Kombination des Funktionsaufbaus eines Neurons, wie sie in Abbildung 3(links) dargestellt ist. Die Transferfunktion des Neurons j (Index n) kann in der Form

$$f_{\text{trans},n j} = f_{\text{aus}}(f_{\text{akt}}(f_{\text{ein}}(\mathbf{o}(t), \mathbf{W})), \theta) \quad (67)$$

mit der Aktivierungsfunktion f_{akt} , der Ausgabefunktion f_{aus} und dem Schwellenwert θ angegeben werden. Mathematisch kennzeichnet der Schwellenwert den Punkt maximaler Steigung und damit Sensitivität der Aktivierungsfunktion f_{akt} . Die in der vorliegenden Arbeit verwendete Standardeingabefunktion des Neurons j bildet die gewichtete Summe nach der Formel

$$f_{\text{ein},n j} = \sum_i w_{i j} o_i . \quad (68)$$

Dabei bezeichnet $o_i(t)$ die Ausgaben der mit Neuron j verbundenen und in Propagierungsrichtung vorher angeordneten Neuronen i (Vorgängerneuronen) zum Iterationszeitpunkt t . $w_{i j}$ kennzeichnet die zugehörigen Gewichte der Verbindungen von den Vorgängerneuronen i zum Neuron j . Die Ausgabe eines verdeckten Neurons oder eines Ausgabeneurons j folgt damit zu

$$o_j(t) = f_{\text{trans},n j}(\mathbf{W}_j, \mathbf{o}(t), \theta) \quad (69)$$

mit $\mathbf{o}(t)$ als dem Vektor der Ausgaben der mit dem Neuron j verbundenen Vorgängerneuronen und \mathbf{W}_j als j -te Spalte der Gewichtsmatrix \mathbf{W} . Ist Neuron j Eingabeneuron, so gilt für die Ausgabe des Neurons j

$$o_j(t) = f_{\text{trans},n j}(\mathbf{in}) . \quad (70)$$

Die Funktionenabfolge in Gleichung (67) liegt fest, kann aber individuell mit verschiedenen Funktionen belegt und so zusammen mit dem Netzplan der Netztopologie an die untersuchte Problemstellung angepaßt werden (siehe Abschnitt 5.2).

Die Abbildung 3 (rechts) zeigt ein Netzwerk im Ausgabeformat des SNNS-Simulators. Die Zahl oberhalb der Neuronen bezeichnet die Nummer des jeweiligen Neurons, die Zahl unterhalb den aktuellen Wert der Ausgabe o des jeweiligen Neurons. Die Neuronen 1 und 2 sind Eingabeneuronen, Neuron 3 bis 5 sind verdeckte Neuronen und Neuron 6 ist ein Ausgabeneuron. Die Gewichtsmatrix \mathbf{W} des Neuronalen Netzes in Abbildung 3 (rechts) kann in der

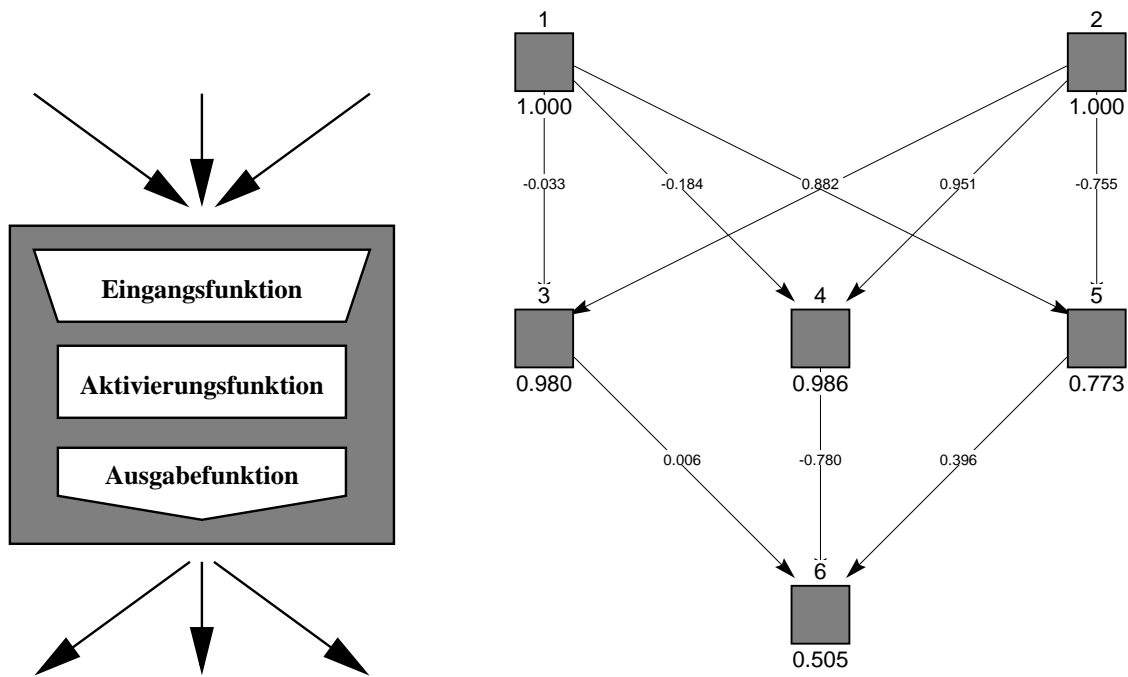


Abbildung 3: Standardneuron (links) und Nomenklatur des SNNS-Simulators (rechts)

Form

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{16} \\ w_{21} & w_{22} & \cdots & w_{26} \\ \vdots & \vdots & \vdots & \vdots \\ w_{61} & w_{62} & \cdots & w_{66} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -0.033 & -0.184 & 0.882 & 0 \\ 0 & 0 & 0.985 & 0.951 & -0.755 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.006 \\ 0 & 0 & 0 & 0 & 0 & -0.780 \\ 0 & 0 & 0 & 0 & 0 & 0.396 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (71)$$

angegeben werden (In Abbildung 3 (rechts) verdeckt das Gewicht $w_{15} = 0.882$ das Gewicht $w_{23} = 0.985$). \mathbf{W} beschreibt die Verbindungen der n_n Neuronen des Netzwerks untereinander und hat die Dimension $(n_n \times n_n)$ (siehe auch Abbildung 4). Die Anzahl der Verbindungen und damit die Anzahl der Verbindungsgewichte im Netzwerk ist n_w . Die Gewichte der Hauptdiagonale und die untere Dreiecksmatrix in \mathbf{W} sind, charakteristisch für ein *Feedforward*-Netzwerk, null, d.h. die zugehörigen Verbindungen existieren nicht (siehe z.B. Gleichung (71)). Die Schreibweise $(n - v - m)$ -Netzwerk mit $n_n = n + v + m$ definiert die Mesostruktur einer Netztopologie mit n Neuronen in der Eingabeschicht, einer verdeckten Schicht mit v verdeckten Neuronen und einer Ausgabeschicht mit m Neuronen wie in Abbildung 4 dargestellt.

Die Trainingsdaten T eines Neuronalen Netzes bestehen aus p Lernvektoren $\mathbf{l}_l = (\mathbf{in}_l, \mathbf{t}_l)^T$ mit $l = 1, \dots, p$, die sich für jeden Trainingsvektor (Muster) l aus dem Eingabevektor \mathbf{in}_l für die n

Eingabeeinheiten

$$\mathbf{in}_l = (in_{l1}, \dots, in_{ln})^T \quad (72)$$

und dem Vektor der Sollausgabe \mathbf{t}_l für die m Ausgabeneinheiten

$$\mathbf{t}_l = (t_{l1}, \dots, t_{lm})^T \quad (73)$$

zusammensetzen. Die Netzübertragungsfunktion als mathematischer Zusammenhang der Ab-

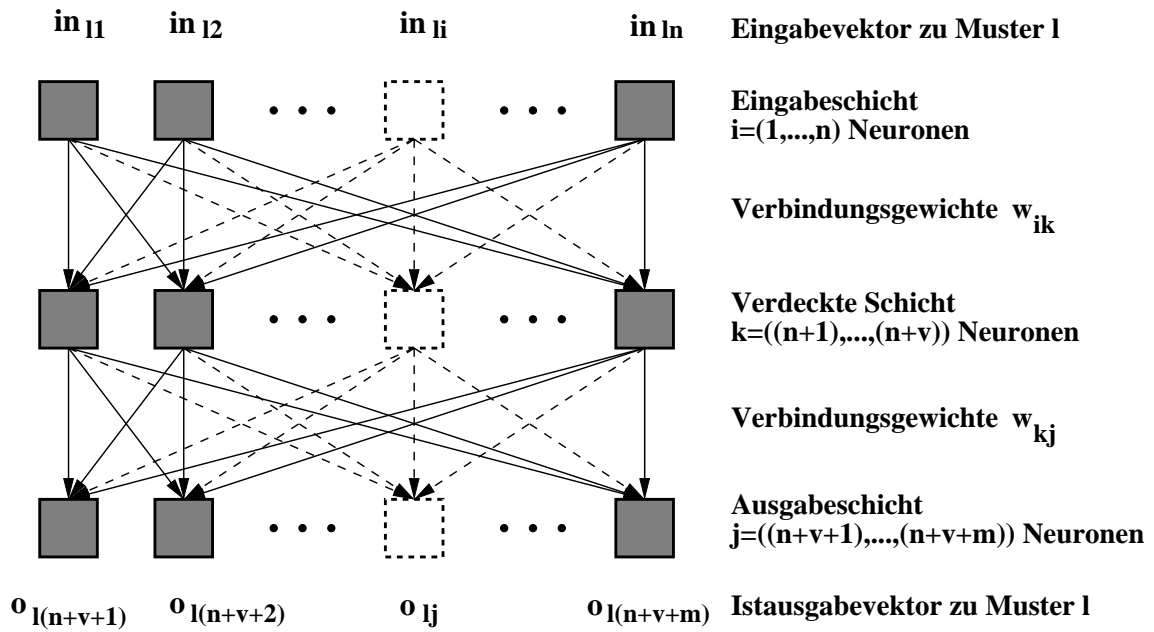


Abbildung 4: Vollständig ebenenweise vernetztes *Feedforward*-Netzwerk

bildung der Netzeingabe auf die Ausgabewerte der Ausgabeneuronen ist durch die Operatoren der Transferfunktionen und den Netzplan der Neuronen zusammen mit der zugehörigen aktuellen Gewichtsmatrix festgelegt. Die Netzübertragungsfunktion $NF_j(t)$ beschreibt den mathematischen Zusammenhang, nach dem das Neuronale Netz im aktuellen Trainings- bzw. *Pruning*-Zustand (siehe Abschnitt 5.5) die n Eingabewerte \mathbf{in}_l des aktuellen Trainingsmusters l gemäß

$$o_{lj}(t) = NF_j(t)(\mathbf{in}_l, \mathbf{W}, \theta) \quad (74)$$

auf den Istausgabewert $o_{lj}(t)$ der Ausgabeneinheit j abbildet. Damit ergibt sich der Vektor der Istausgaben zugehörig zum Eingabevektor \mathbf{in}_l zu

$$\mathbf{o}_l = (o_{l(n+v+1)}, \dots, o_{l(n+v+m)})^T \quad (75)$$

5.2.2 Konventionelle Neuronale Netze

Der Begriff konventionelle Neuronale Netze bezeichnet ebenenweise vollständig verbundene *Feedforward*-Netzwerke mit sigmoiden (S-förmigen) Aktivierungs- bzw. Transferfunktionen. Bei den verwendeten sigmoiden Aktivierungsfunktionen handelt es sich um die logistische Aktivierungsfunktion

$$f_{\text{akt},1}(x) = \frac{1}{1 + e^{-x}} \quad (76)$$

sowie den Tangens hyperbolicus

$$f_{\text{akt},2}(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (77)$$

Beide Funktionen sind stetig, monoton steigend, differenzierbar und haben als Wertebereiche das offene Intervall $]0, 1[$ bzw. $] - 1, 1[$. Zusammen mit einer linearen Ausgabefunktion $f_{\text{aus}} = x$ ergibt sich die Ausgabe des Neurons j im Iterationszeitpunkt $(t+1)$ mit $f_{\text{akt},k}$ als Aktivierungsfunktion des Neurons j und $k = 1$ oder 2 gemäß Gleichung (76) oder (77) zu

$$o_j(t+1) = f_{\text{akt},k} \left(\sum_i w_{ij} o_i(t) + \theta_j(t) \right). \quad (78)$$

Die verwendeten sigmoiden Aktivierungsfunktionen zeigen asymptotisches Verhalten außerhalb eines schmalen Intervalls um den Schwellenwert θ . Als Folge ist das Veränderungspotential der Neuronenaktivierung und damit ihre Lernfähigkeit stark eingeschränkt. Ein sinnvoller Definitionsbereich der jeweiligen Aktivierungsfunktion ist außerhalb der Sättigungsbereiche zu suchen. Die Test- und Trainingsdaten konventioneller Neuronaler Netze werden deshalb auf den sinnvollen Definitionsbereich der Aktivierungsfunktionen $]0.05, 0.95[$ bzw. $] - 0.95, 0.95[$ transformiert. Es ist sicherzustellen, daß Test- und Trainingsdaten mit identischen Transformationsgrenzen normiert werden.

Die sich aus einer solchen Transformation ergebenden Datensysteme aus Test- und Trainingsdaten sind numerisch ausgeglichen. Der Informationsgehalt zur Problemstellung ist aufgrund des engen Normierungsbereichs aber auf die Dezimalstellen der Muster beschränkt. Daraus resultiert eine Abhängigkeit der Trainings- und Testergebnisse von der Rechengenauigkeit der Simulationssoftware.

Die im SNNS implementierte Funktionalität der Neuronen der Eingabeschicht ist eingeschränkt. Die Aktivierungsfunktion der Eingabeneuronen hat keinen Einfluß auf die Transferfunktion der Eingabeneuronen. Deren individuelle Funktionalität kann statt dessen über die Ausgabefunktion berücksichtigt werden (siehe Neuron $i = 1, 2$ in Gleichung (79)).

Die Netzübertragungsfunktionen konventioneller Neuronaler Netze sind im allgemeinen sehr unübersichtlich. Als Beispiel sei die Netzübertragungsfunktion des einfachen, konventionel-

len Netzwerks in Abbildung 3(rechts) genannt. Für die Transferfunktionen aller Neuronen seien die Angaben aus Gleichung (78) gültig. Damit folgt unter der Annahme $\theta = 0$ mit $\mathbf{in}_{l,i}$ als Netzeingabe zum Eingabeneuron i und Muster l sowie $f_{\text{trans},n i}$ als Transferfunktion des Neurons i die Netzübertragungsfunktion zu

$$\begin{aligned} \text{NF}_6(\mathbf{in}, \mathbf{W}) = & f_{\text{trans},n 6} \left[w_{3 6} f_{\text{trans},n 3} \left(\sum_{i=1}^2 w_{i 3} f_{\text{aus},n i}(\mathbf{in}_{l,i}) \right) \right. \\ & + w_{4 6} f_{\text{trans},n 4} \left(\sum_{i=1}^2 w_{i 4} f_{\text{aus},n i}(\mathbf{in}_{l,i}) \right) \\ & \left. + w_{5 6} f_{\text{trans},n 5} \left(\sum_{i=1}^2 w_{i 5} f_{\text{aus},n i}(\mathbf{in}_{l,i}) \right) \right]. \end{aligned} \quad (79)$$

Die Anforderungen der untersuchten Problemstellung legen die Anzahl notwendiger Ein- und Ausgabeneinheiten fest. Bei der Festlegung der Anzahl verdeckter Schichten und Neuronen pro verdeckter Schicht ist zwischen Über- und Unterdimensionierung abzuwägen. Unterdimensionierung der Netztopologie kann bei unbefriedigendem Lernverhalten zu Oszillation und Stagnation des Netzfehlers auf hohem Niveau führen, während bei Überdimensionierung das Netz die Stützstellen der Trainingsdaten u.U. exakt zu repräsentieren lernt und die Abstraktionsfähigkeit auf unbekannte Testdaten leidet.

Das Vorhersagepotential konventioneller Neuronaler Netze bezüglich unbekannter Testdaten ist auf die Wertebereiche der im Lernverfahren eingesetzten Trainingsdaten begrenzt. Man spricht von beschränkter Generalisierungsfähigkeit.

Für konventionelle Neuronale Netze existieren Heuristiken, die aber nur Anhaltspunkte zur Festlegung der Anzahl von Neuronen in den verdeckten Schichten geben (siehe z.B. VYŠNIAUSKAS ET AL. [105]). Mindestanforderungen hinsichtlich Aufbau und Funktionalität einer Netztopologie zur Approximation von Zusammenhängen finden sich u.a. bei HORNIK UND STINCHCOMBE [106], STINCHCOMBE [107, 108] sowie CHEN ET AL. [109].

5.2.3 Problemspezifische Neuronale Netze

Der Begriff „Problemspezifische Neuronale Netze“ kennzeichnet Netzwerke, bei denen bestehendes Wissen zur Problemstellung vor Beginn des eigentlichen Netztrainings in die Netztopologie implementiert wird. Als Beispiel sei im folgenden die identische Abbildung einer mathematischen Ansatzfunktion in eine Ausgangsnetztopologie (ANT) beschrieben. Die verwendete Systematik fundiert auf grundlegenden Untersuchungen von BARTH [8], EMRICH [9], HERRMANN [10, 110], PAO [5], RUDOLPH [11, 12, 13, 14] sowie RUDOLPH UND KRÖPLIN [16]. In diesen Arbeiten wird auch eine Darstellung von Anwendungsmöglichkeiten

der Dimensionsanalyse u.a. in der Strukturmechanik gegeben, auf die die vorliegende Arbeit aufbaut.

Die Bestimmung der variablen Parameter der mathematischen Ansatzfunktion bzw. der äquivalenten Netzübertragungsfunktion der Ausgangsnetztopologie erfolgt über Training und *Pruning* der Ausgangsnetztopologie. Die variablen Parameter umfassen dabei die veränderlichen Verbindungsgewichte und in einem erweiterten Sinn hinsichtlich *Pruning* auch die Existenz von Verbindungen und verdeckten Neuronen (siehe Abschnitt 5.5). In Erweiterung zu der

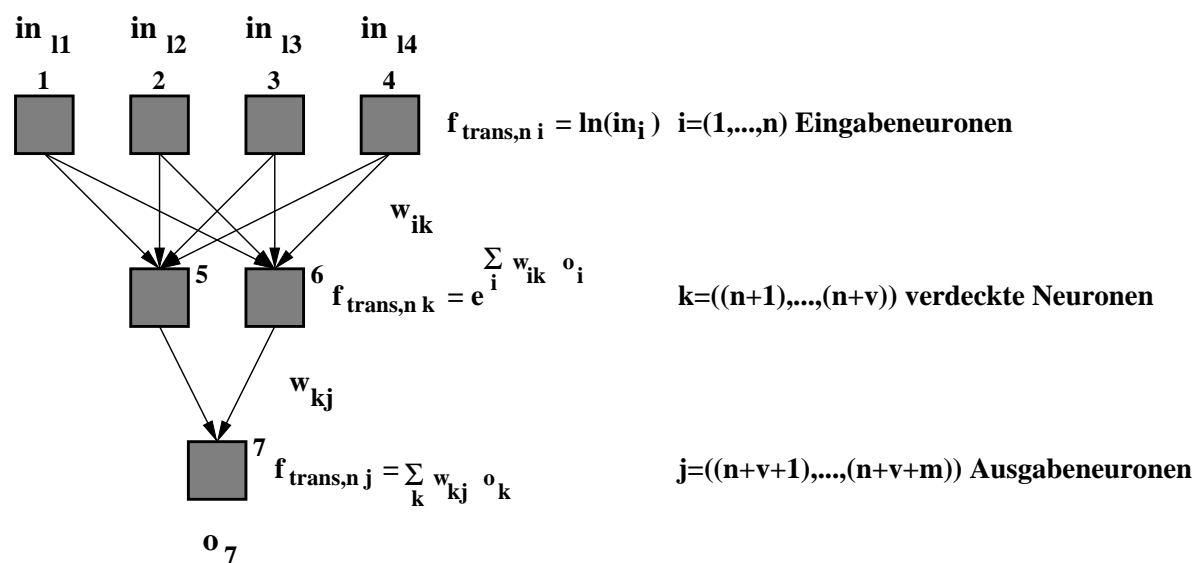


Abbildung 5: Problemspezifische Netztopologie: Summe zweier Produktfunktionen

in den obigen Zitatstellen beschriebenen Systematik demonstrieren die nachfolgenden Formulierungen die identische Abbildung eines erweiterten Produktansatzes in ein Neuronales Netz. Die Ausführungen beziehen sich dabei auf die Netztopologie in Abbildung 5 mit $n = 4, v = 2, m = 1$ und gelten bzgl. des Eingabevektors in_1 für ein beliebiges Trainingsmuster l , dessen Erwähnung zur Vereinfachung der Schreibweise ebenso wie der aktuelle Iterationszeitpunkt t weggelassen wird. Die Transferfunktion der Eingabeneuronen $i = 1, \dots, (n = 4)$ sei der natürliche Logarithmus, realisiert in der Ausgabefunktion der Neuronen i . Die Transferfunktion der verdeckten Neuronen $k = 5, 6$ sei die Exponentialfunktion zur Basis e und die Transferfunktion des Ausgabeneurons $j = 7$ sei die Identität. Für alle Neuronen gelte die Standardeingabefunktion nach Gleichung (68). Der Schwellenwert θ bleibt hier unberücksichtigt. Eine mathematische Darstellung der Transferfunktionen der drei Netzschichten folgt zu

$$o_i = f_{trans,n i}(in_i) = \ln(in_i) \text{ für alle } i \in \{1, \dots, (n = 4)\}$$

$$o_k = f_{\text{trans},n\ k}(\mathbf{W}, \mathbf{in}) = e^{\sum_{i=1}^4 w_{i\ k} o_i} = \prod_{i=1}^4 in_i^{w_{i\ k}} \text{ für alle } k \in \{5, 6\} \quad (80)$$

$$o_{j=7} = f_{\text{trans},n\ j=7}(\mathbf{W}, \mathbf{in}) = \sum_{k=5}^6 w_{k\ 7} o_k .$$

Für die Netztopologie in Abbildung 5 ergibt sich die Netzübertragungsfunktion NF zu

$$o_7 = \text{NF}(\mathbf{W}, \mathbf{in}) = w_{5\ 7} \prod_{i=1}^4 in_i^{w_{i\ 5}} + w_{6\ 7} \prod_{i=1}^4 in_i^{w_{i\ 6}} . \quad (81)$$

Damit ist der erweiterte Produktansatz aus Gleichung (81) durch eine Kombination aus individueller Neuronenfunktionalität, problemspezifischer Netzstruktur sowie einem geeigneten Netzplan mit zugehöriger Belegung der Matrix der Verbindungsgewichte identisch in der Netztopologie in Abbildung 5 realisiert. Dies gelingt durch Abbildung der Matrix der Verbindungsgewichte der trainierten bzw. durch *Pruning* reduzierten Netztopologie auf die unbekannt Parameter des Funktionsansatzes und setzt eine erfolgreiche Erfassung des den Trainingsmustern inhärenten Zusammenhangs durch die Netztopologie voraus. Weitere mögliche Ansatzfunktionen sind Summen- bzw. Polynomansätze. Es sei auf die zitierte Literatur und ein Beispiel in Abschnitt 5.5.1 verwiesen.

Vorteil der Systematik ist die Möglichkeit, vorhandenes Wissen zur untersuchten Problemstellung vor Beginn der Trainings- bzw. *Pruning*-Phase in die Ausgangsnetztopologie zu implementieren. Die variablen Parameter der mathematischen Ansatzfunktion sind durch die beschriebene Methodik dezidierten Verbindungsgewichten $w_{i\ j}$ der Ausgangsnetztopologie zugeordnet. Die Bestimmung der Variablen erfolgt über Trainings- und *Pruning*-Verfahren als Resultat auf den Zusammenhang, der inhärent in den Mustern der Trainingsdaten enthalten ist. Eine Regression des den Trainingsdaten inhärenten Zusammenhangs unter der Voraussetzung erfolgreichen Trainings bzw. *Prunings* ist möglich, falls die in die Netztopologie abgebildete Ansatzfunktion in der Lage ist, diesen Zusammenhang mathematisch vollständig zu erfassen. Damit besitzt die trainierte und evtl. durch *Pruning* reduzierte Netztopologie die maximale Abstraktionsfähigkeit zur untersuchten Problemstellung, da die Netzfunktion den mathematischen Zusammenhang im gesamten Gültigkeitsbereich der zugrundeliegenden physikalischen Modellbildung exakt erfaßt.

Mit der Auswahl einer problemspezifischen Netztopologie, die zur Repräsentation der aktuellen Problemstellung verwendet werden soll, schränkt man gleichzeitig aber auch das Beschreibungspotential der Netztopologie bzw. der zugehörigen Netzübertragungsfunktion ein (siehe Abschnitt 5.5.1). Diese Einschränkung kann, sollte sie falsch sein oder die Darstellung des in den Trainingsdaten enthaltenen funktionalen Zusammenhangs potentiell nicht ermöglichen, eine exakte Repräsentation des gesuchten Zusammenhangs verhindern.

5.2.4 Multivariabler Summen-Produkt-Ansatz

Während in Abschnitt 5.2.3 die Abbildung spezieller mathematischer Funktionen in ein Neuronales Netz dargestellt ist, wird im vorliegenden Abschnitt darauf aufbauend die Entwicklung einer Kombination zweier problemspezifischer Ansätze vorgestellt. Diese sollen das mathematische Erfassungspotential der Netzübertragungsfunktion der Ausgangsnetztopologie erweitern und speziell bei der Bestimmung problemspezifischer Netzwerke aus der Reduzierung (*Pruning*) allgemeiner Ausgangsnetztopologien eingesetzt werden (siehe Abschnitt 5.5). Ziel ist es, mit minimalen Vorabkenntnissen zur Problemstellung Netztopologien zu ermitteln, die in der Lage sind, den zugrundeliegenden physikalischen Vorgang der Problemstellung aus den Trainingsdaten exakt zu erfassen. Diese haben dann eine maximale Generalisierungsfähigkeit in dem Sinne, daß das Vorhersagepotential des trainierten Netzwerks nicht auf die Grenzen des Intervalls der Trainingsmuster beschränkt ist.

Das erfolgreich trainierte problemspezifische Netzwerk bildet statt dessen unbekannte Eingabedaten zur Problemstellung über dem gesamten Definitionsbereich des beschriebenen physikalischen Vorgangs auf die korrekte Lösung ab. Dies kann als eine Form der Datenkomprimierung angesehen werden, da der inhärente funktionale Zusammenhang der Trainingsdaten vollständig durch das trainierte Netzwerk repräsentiert ist. Als Beispiel sei die analytisch exakte Ermittlung des Verschiebungsverhaltens sämtlicher Knoten eines Fachwerks genannt (siehe Abschnitt 6.4).

Die eingesetzte Systematik nutzt die in Abschnitt 5.2.3 beschriebenen Methoden zur identischen Abbildung einer mathematischen Funktion in ein Neuronales Netz. Durch Aufsummierung mehrerer Produktansätze über v verdeckte Neuronen für m Ausgabeneuronen ergibt sich für ein $(n - v - m)$ -Netzwerk mit Eingabevektor \mathbf{in}_l und dem Vektor der Istausgabe \mathbf{o}_l ein erweiterter Produktansatz zum Trainingsmuster l in allgemeiner Form zu

$$\mathbf{o}_{lj} = \text{NF}_j(\mathbf{in}, \mathbf{W}) = \sum_{k=n+1}^v \left(w_{kj} \prod_{i=1}^n \text{in}_{li}^{w_{ik}} \right) \quad \text{für alle } j \in \{(n + v + 1), \dots, (n + v + m)\} . \quad (82)$$

Gleichung (82) beschreibt die Netzübertragungsfunktion für die m Ausgabeneuronen eines problemspezifischen Neuronalen Netzes mit multivariablen Summen-Produkt-Ansatz (siehe Abbildung 6). Die Netzübertragungsfunktion kann individuell als Funktionsansatz eines problemspezifischen Neuronalen Netzes dem zu erfassenden Zusammenhang angepaßt werden. Hierin liegt der Hauptvorteil gegenüber konventionellen Neuronalen Netzen.

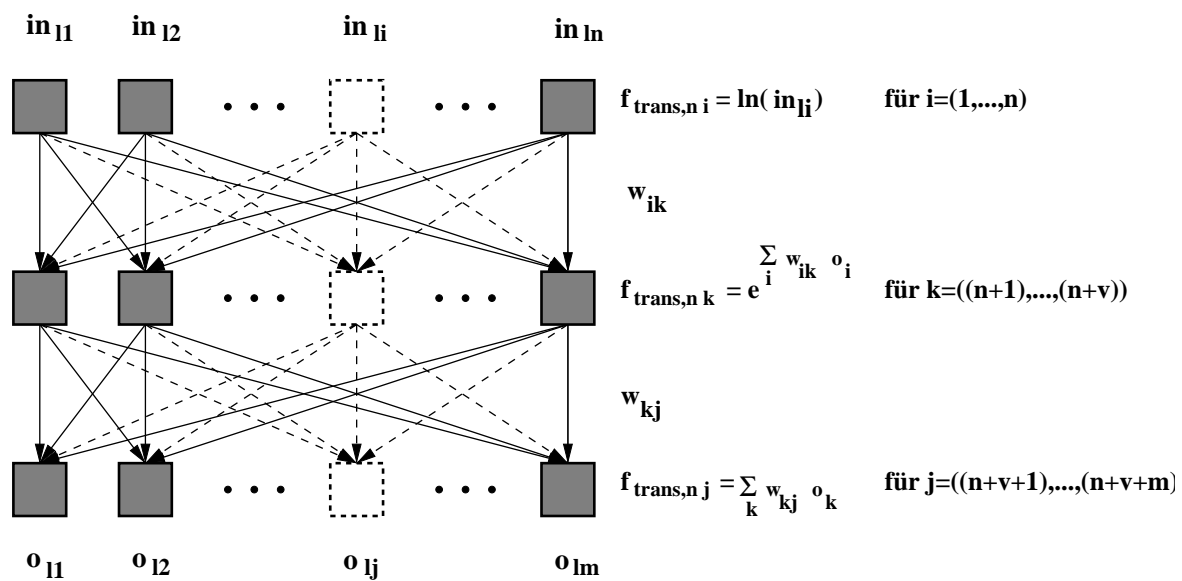


Abbildung 6: Problemspezifisches Neuronales Netz mit multivariablen Summen-Produkt-Ansatz

5.3 Test- und Trainingsdaten

Für die durchgeführten Untersuchungen statisch belasteter Fachwerkstrukturen kann die Erzeugung der Test- und Trainingsdatensysteme experimentell oder über Simulationsrechnungen erfolgen. Aufgrund des Umfangs und der notwendigen Variationsbreite der Datensysteme sowie aus Zeit- und Kostengründen werden in der vorliegenden Arbeit sämtliche Test- und Trainingsdaten durch Simulation des Verhaltens der untersuchten Struktur erzeugt.

Das hierzu eingesetzte FEM-Programm FEA ist eine Weiterentwicklung des FEM I Grundprogramms nach AHRENS UND DINKLER [111]. Weitere notwendige Verfahren zur Erzeugung von Test- und Trainingsdaten umfassen einen Permutationsalgorithmus zur Steuerung des FEM-Programms FEA sowie das kombinierte FEM-Zuverlässigkeits-Programm FEMZA (siehe Abschnitt 4.5). Zusätzlich entwickelte Algorithmen werden zur Vorverarbeitung der Rohtrainingsdaten mittels Dimensionsanalyse und linearer Transformation eingesetzt. Der Begriff Rohtrainingsdaten bezeichnet ausgewählte Parameter, die den aktuellen Zustand der untersuchten Fachwerkstruktur beschreiben (z.B. Knotenlasten, Steifigkeiten bzw. Elastizitäten) sowie ausgewählte Strukturantworten aus den Simulationsrechnungen mit FEA und FEMZA vor einer Datenvorverarbeitung.

5.3.1 Erzeugung von Test- und Trainingsdaten

Die Auswahl signifikanter Kenngrößen der untersuchten Problemstellung legt die Ein- und Ausgabemuster des Netzwerks fest. Für problemspezifische Netze wird diese Auswahl dann in der mathematischen Ansatzfunktion repräsentiert und in die Netztopologie abgebildet. In konventionellen Netzwerken bzw. bei unbekanntem Zusammenhängen sind die Ein- und Ausgabedaten durch die verfügbaren Informationen zur Problemstellung sowie durch Limitierungen hinsichtlich des zulässigen Rechenaufwands gegeben. Die Auswahl der Ausgabedaten ist durch die verfügbaren Systemreaktionen und die Bedürfnisse der durch das Netzwerk zu beschreibenden Ausgabeinformationen gegeben.

Als Beispiel sei die Schadensanalyse statisch bestimmter Fachwerke in Abschnitt 8.3 genannt. Bei den dort eingesetzten konventionellen Netzwerken werden Informationen zur Systemantwort (Knotenverschiebungen) als Eingabe- und die gesuchten Integritäten der Stäbe als Ausgabegrößen verwendet.

Die Festlegung der Netztopologie aus der gewählten mathematischen Methodik des Netzansatzes bestimmt die notwendige Datenvorverarbeitung der Rohtrainingsdaten vor Trainingsbeginn (siehe Abschnitt 5.3.3 und 6.1.1). Dies ist gekoppelt mit der Erzeugung der Lernvektoren für Test und Training.

Nach Abschluß des Lernvorgangs mit Hilfe der Trainingsdaten ermöglichen die dem Netz bis dahin unbekanntem Testdaten eine Kontrolle des Lernerfolgs und der Generalisierungsfähigkeit des trainierten Netzes (*Recall*). Test- und Trainingsdaten beschreiben Parameterkombinationen zu ein und derselben Problemstellung. Die Parameter der Testdaten konventioneller Netzwerke sollten normalerweise im Inneren des Parameterbereichs der Trainingsdaten liegen, da die Generalisierungsfähigkeit konventioneller Neuronaler Netze auf den trainierten Parameterbereich beschränkt ist. Das Testergebnis ist von der Wahl der Testdaten abhängig. Der gesamte Ablauf von der Auswahl der in den Test- und Trainingsmustern zu repräsentierenden Kenngrößen der Problemstellung bis zur statistischen Auswertung der Testergebnisse ist schematisch in Abbildung 7 dargestellt. In der vorliegenden Arbeit sind Muster, die stochastische Vorgänge beschreiben, ausgeschlossen.

5.3.2 Permutationsalgorithmus

Der Permutationsalgorithmus des Programms FEMZA steuert die äquidistante Parametervariation der Eingabedatei des kombinierten FEM-Zuverlässigkeits-Programms FEMZA bei der Erzeugung von Rohtrainingsdaten für die Beschreibung des Zuverlässigkeitsverhaltens von

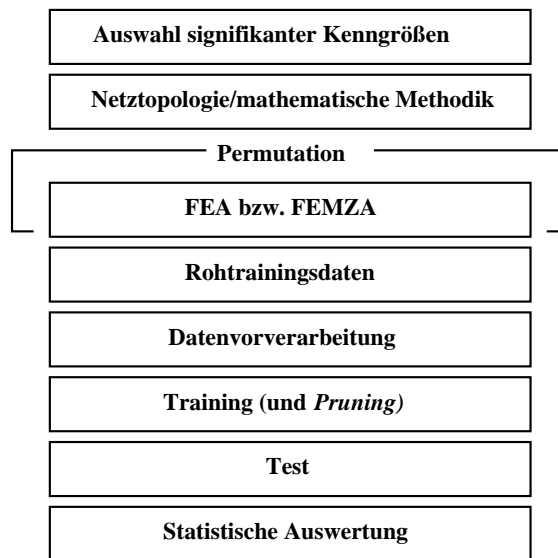


Abbildung 7: Algorithmus zur Auswahl und Erzeugung der Test- und Trainingsdaten

Strukturen. Der Permutationsalgorithmus des Programms FEA steuert die äquidistante Parametervariation der Eingabedatei des FEM-Programms FEA. Dies dient der Generierung von Systemantworten statisch belasteter Strukturen z.B. zur Erzeugung von Rohtrainingsdaten für die Analyse geschädigter Strukturen (siehe Kapitel 8).

So können alle Kombinationsmöglichkeiten ausgewählter Parameter einer oder mehrerer Parametergruppen bei jeweils individuell festgelegter, äquidistanter Schrittweite ermittelt und zu jeder der Kombinationsmöglichkeiten über eine Simulationsrechnung mit FEA bzw. FEMZA die zugehörige Systemantwort bestimmt werden. Für ein Fachwerk mit s Stäben und k gleichzeitig wirkenden Knotenlasten ergeben sich bei der Variation aller Stabdurchmesser um v_1 Inkremente und Variation aller Knotenlasten um v_2 Inkremente ($v_1^s \cdot v_2^k$) verschiedene Parameterkombinationen. Aus Strukturparametern und zugehöriger Systemantwort ergeben sich über die Datenvorverarbeitung der Rohtrainingsdaten die Lernvektoren der Trainingsdatensysteme.

5.3.3 Datenvorverarbeitung durch lineare Transformation

Die Datenvorverarbeitung der Rohtrainingsdaten durch lineare Transformation oder Dimensionsanalyse vor Beginn des Netztrainings garantiert numerisch ausgeglichene Test- und Trainingsdaten. Sollten z.B. Elemente des Eingabevektors um mehrere Größenordnungen verschieden sein, so kann dies den Lernerfolg behindern und zu Stagnationen des Netzfehlers in lokalen Minima der Hyperfläche des Gesamtnetzfehlers führen (siehe Abschnitt 5.6.2). Bei

der Datenvorverarbeitung der Rohtrainingsdaten ist zwischen Datensystemen für konventionelle und problemspezifische Neuronale Netze zu unterscheiden. Die Datenvorverarbeitung durch Dimensionsanalyse ist in Abschnitt 6.1.1 beschrieben.

Die Notwendigkeit einer linearen Transformation der Test- und Trainingsdaten für konventionelle Neuronale Netze (siehe Abschnitt 5.2.2) ist durch das asymptotische Verhalten der eingesetzten sigmoiden Aktivierungs- bzw. Transferfunktionen gegeben. Beide Funktionen (siehe Gleichung 76 und 77) zeigen für große negative und positive Eingabeargumente einen asymptotischen Verlauf der Funktionswerte gegen 0 und 1 bzw. gegen -1 und 1. Als Resultat würde ein weiter Bereich positiver oder negativer Argumente entsprechender Transferfunktionen Gradienten nahe null zeigen. Dies würde den Lernerfolg deutlich beeinträchtigen oder gar verhindern und wird durch die Normierung der Ein- und Ausgabevektoren der Rohtrainingsdaten auf sinnvolle Normierungsbereiche wie $]0.05, 0.95[$ bzw. $] - 0.95, 0.95[$ ausgeschlossen.

Die hierzu verwendete lineare Transformation kann für das i -te Element des Musters l rin_{li} des Eingabevektors der Rohtrainingsdaten \mathbf{rin}_l für den Normierungsbereich $]a, b[$ in der Form

$$in_{li} = \frac{(b - a)}{(rin_{li,max} - rin_{li,min})} (rin_{li} - rin_{li,min}) + a \quad (83)$$

mit $rin_{li,min}$ bzw. $rin_{li,max}$ als minimal bzw. maximal auftretendem Wert des Elements i aller $l = 1, \dots, p$ Eingabevektoren der Rohtrainingsdaten angegeben werden. Die Transformation der Ausgabevektoren funktioniert analog.

Der Einfluß der verwendeten Normierungsbereiche $]0.05, 0.95[$ und $] - 0.95, 0.95[$ sowie ihr Zusammenwirken mit den Transferfunktionen der Neuronen für $k = 1, 2$ in Gleichung (78) wird im Rahmen der Untersuchungen zur Schadensanalyse von Fachwerkstrukturen in Kapitel 8 diskutiert.

5.4 Netztraining

Überwachte Lernverfahren für Neuronale Netze sind Algorithmen, die die Variablen der Netztopologie während der wiederholten Präsentation der Trainingsdaten so verändern, daß die Gesamtfehlerfunktion aus der Abweichung zwischen Istausgabe des Netzwerks und zugehöriger Sollausgabe der Trainingsdaten ein Minimum annimmt. Der gewählte Lernalgorithmus verändert mit Hilfe der Lernvektoren der Trainingsdaten die variablen Netzgewichte und Schwellenwerte sowie bei zusätzlichem *Pruning* auch die Anzahl der verdeckten Neuronen und den zugehörigen Netzplan der *links*.

Im folgenden werden das generelle Vorgehen bei Training und Test Neuronaler Netze erläutert

sowie die verwendeten deterministischen Lernalgorithmen *Backpropagation with Momentum Term and Flat Spot Elimination* und *Resilient Propagation* besprochen.

5.4.1 Training und Test

Vor dem Start des Lernalgorithmus werden die Gewichte des Netzwerks mit gleichverteilten Zufallswerten aus einem Intervall $[b, a]$ initialisiert, um eine Lernblockade durch *symmetry breaking* zu verhindern (siehe z.B. ZELL [23]).

Lernverfahren verändern während einer diskreten Anzahl von Lernepochen die Variablen des Netzwerks im Sinne einer Minimierung der Differenz zwischen Soll- und Istausgabevektor. Ein Trainingszyklus (Epoche E_p) des Lernverfahrens besteht aus der einmaligen Vorwärtspropagierung der Eingabevektoren \mathbf{in}_l zur Ausgabe- und der zugehörigen Rückpropagierung des Fehlersignals von der Ausgabe- zur Eingabeschicht für alle $l = 1, \dots, p$ Lernvektoren $\mathbf{l}_l = (\mathbf{in}_l, \mathbf{t}_l)^T$ der Trainingsdaten. Die Eingabevektoren der p Trainingsmuster werden einzeln und in zufälliger Reihenfolge an die Eingabeschicht des Netzwerks angelegt und durch das Netz propagiert. Die aktuelle Parametereinstellung des Netzes bildet die \mathbf{in}_l dann jeweils auf die zugehörige Istausgabe o_{lj} ab. Das Fehlersignal ergibt sich aus der Differenz zwischen der gewünschten Sollausgabe (*teaching output*) der Trainingsdaten t_{lj} und der tatsächlichen Istausgabe o_{lj} , jeweils zugehörig zum Trainingsmuster l und Ausgabeneuron j .

Die Rückwärtspropagierung des Fehlersignals gemäß der Vorschrift des Lernalgorithmus dient der Änderung der Verbindungsgewichte im Sinne einer Verringerung des Gesamtnetzfehlers. Die Gesamtfehlerfunktion eines Neuronalen Netzes mit Ausgabeneuronen $j = (n + v + 1), \dots, (n + v + m)$ und p Trainingsmustern $l = 1, \dots, p$ ist durch

$$E = \sum_{l=1}^p \left[\frac{1}{2} \sum_{j=(n+v+1)}^{(n+v+m)} (t_{lj} - o_{lj})^2 \right] \quad (84)$$

gegeben. Der jedem Neuron zugeordnete Schwellenwert θ wird analog zu den Verbindungsgewichten durch den Lernalgorithmus verändert. Der Neuronale Netze Simulator SNNS verwendet das Fehlermaß SSE (*Sum Squared Error*) in der Form

$$\text{SSE} = \sum_{l=1}^p \left[\sum_{j=(n+v+1)}^{(n+v+m)} (t_{lj} - o_{lj})^2 \right]. \quad (85)$$

Die Testphase eines Neuronalen Netzes ermöglicht eine Kontrolle des Lernerfolgs und der Generalisierungsfähigkeit des trainierten Netzwerks gegenüber unbekanntem Testdaten (*Recall*). Dies geschieht durch Abbildung der Eingabevektoren der Testdaten \mathbf{tin}_l auf die Istausgabevektoren \mathbf{to}_l über den aktuellen Zustand (t) der Netztopologie und Gewichtsmatrix. Der Parameter t kennzeichnet den aktuellen Trainingszustand der Netztopologie.

Zur statistischen Auswertung der Testergebnisse ist der Mittelwert des Gesamttestfehlers μ_{test} aus dem Vergleich von Istausgabe \mathbf{to}_l und Sollausgabe \mathbf{tt}_l der Testdaten mit den Ausgabeneuronen $j = (n + v + 1), \dots, (n + v + m)$ für tp Testmustern $l = 1, \dots, tp$ zu

$$\mu_{\text{test}} = \frac{1}{tp \cdot m} \sum_{l=1}^{tp} \sum_{j=(n+v+1)}^{(n+v+m)} \frac{|\mathbf{tt}_{lj} - \mathbf{to}_{lj}|}{|\mathbf{tt}_{lj}|} \quad (86)$$

definiert. Die Varianz des Gesamttestfehlers folgt zu

$$\text{Var}_{\text{test}} = \frac{1}{tp \cdot m} \sum_{l=1}^{tp} \sum_{j=(n+v+1)}^{(n+v+m)} \left(\left[\frac{|\mathbf{tt}_{lj} - \mathbf{to}_{lj}|}{|\mathbf{tt}_{lj}|} \right]^2 - [\mu_{\text{test}}]^2 \right). \quad (87)$$

5.4.2 Backpropagation with Momentum Term and Flat Spot Elimination

Bei dem im weiteren kurz BackpropMomentum genannten Algorithmus handelt es sich um ein erweitertes Gradientenabstiegsverfahren, das die Netzgewichte proportional zum negativen Gradienten der Fehlerfunktion E verändert. BackpropMomentum ist ein *online*-Trainingsverfahren, das die Gewichte unmittelbar nach der Präsentation jedes einzelnen Musters l aktualisiert. Es basiert u.a. auf Grundlagen von RUMELHART UND MCCLELLAND [112]. Wichtige Erweiterungen gegenüber einem konventionellen Gradientenverfahren umfassen:

- Gradientenverfahren können in Bereichen kleiner Gradienten der untersuchten Zielfunktion stagnieren und in Bereichen großer Gradientenänderungen oszillieren. Die Addition eines *Momentum*-Terms $\Delta_l w_{ij}(t)$ zur Berücksichtigung der Gewichtsänderung im vorhergehenden Iterationsschritt t bewirkt eine Vergrößerung der Gewichtsänderung bei kleinen Gradienten (flache Plateaus der Hyperfläche der Gesamtfehlerfunktion) und eine Verringerung der Gewichtsänderung im Bereich großer Gradientenänderungen (Oszillation in steilen Tälern der Hyperfläche der Gesamtfehlerfunktion).
- Die Ableitungen der verwendeten sigmoiden Aktivierungsfunktionen streben im Sättigungsbereich der Aktivierungsfunktion gegen null (*flat spots*). Dies hemmt die Anpassung der Gewichte signifikant. Abhilfe leistet die Addition einer Konstante (z.B. $c = 0.1$) zur Ableitung der Aktivierungsfunktion $\frac{\partial}{\partial f_{\text{ein},lk}} f_{\text{akt}}(f_{\text{ein},lk})$.

Für *Feedforward*-Netze ermöglicht die folgende mathematische Formulierung die rekursive Berechnung der Änderung der Verbindungsgewichte zwischen in Propagierungsrichtung vorher angeordneten Neuronen i und nachfolgenden Neuronen j . Sollte j ein verdecktes Neuron sein, so werden in Propagierungsrichtung nachfolgende Neuronen mit k indiziert. Falls

j ein verdecktes Neuron zugehörig zu der in der Arbeit verwendeten Netzeingabefunktion $f_{\text{ein},lj} = \sum_i o_{li} w_{ij}$ ist, ergibt sich zu Muster l die Änderung des Verbindungsgewichts zwischen den Neuronen i und j zu

$$\Delta_1 w_{ij}(t+1) = \eta \delta_{lj} o_{ij} + \alpha \Delta_1 w_{ij}(t) . \quad (88)$$

Das Fehlersignal δ_{lj} ist durch

$$\delta_{lj} = \begin{cases} \left(\frac{\partial}{\partial f_{\text{ein},lj}} f_{\text{akt}}(f_{\text{ein},lj}) + c \right) (t_{lj} - o_{lj}) & j \text{ ist Ausgabeneuron} \\ \left(\frac{\partial}{\partial f_{\text{ein},lj}} f_{\text{akt}}(f_{\text{ein},lj}) + c \right) \left(\sum_k \delta_{lk} w_{jk} \right) & j \text{ ist verdecktes Neuron} \end{cases} \quad (89)$$

definiert. Die vier Steuerparameter des Lernverfahrens BackpropMomentum im Simulator SNNS sind η als Proportionalitätsfaktor des Gradienten α , der den additiv berücksichtigten Anteil an der vorhergehenden Gewichtsänderung $\Delta_1 w_{ij}(t)$ (Momentum-Term) gewichtet, c als additive Konstante zur Ableitung der Aktivierungsfunktion und d_{max} als maximal zulässige Differenz $(t_{lj} - o_{lj})$.

5.4.3 Resilient Propagation

Die Grundidee des auf Arbeiten von RIEDMILLER [113, 114] basierenden *Resilient Propagation*-Verfahrens (Rprop) ist, negative Einflüsse von $\frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t)$ auf die Adaptation der Verbindungsgewichte zu vermeiden. $\frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t)$ bezeichnet die partiellen Ableitungen der erweiterten Gesamtfehlerfunktion E_{erw} nach den Verbindungsgewichten w_{ij} zum Iterationszeitpunkt t . Rprop verwendet nur das Vorzeichen der partiellen Ableitung zur Bestimmung der individuellen Richtung der Gewichtsänderung. Diese ist durch

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t) & \text{falls } \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t) > 0 \\ +\Delta_{ij}(t) & \text{falls } \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t) < 0 \\ 0 & \text{sonst} \end{cases} \quad (90)$$

gegeben. Die Schrittweite der Änderung der Verbindungsgewichte ist ausschließlich durch einen für jedes Gewicht individuellen Änderungswert

$$\Delta_{ij}(t) = \begin{cases} \eta^+ \Delta_{ij}(t-1) & \text{falls } \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t-1) \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t) > 0 \\ \eta^- \Delta_{ij}(t-1) & \text{falls } \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t-1) \frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t) < 0 \\ \Delta_{ij}(t-1) & \text{sonst} \end{cases} \quad (91)$$

festgelegt. Jeder Vorzeichenwechsel der partiellen Ableitung $\frac{\partial E_{\text{erw}}}{\partial w_{ij}}(t)$ wird als zu weit gehender Sprung über ein lokales Minimum der Gesamtfehlerfunktion interpretiert und der Betrag

der Gewichtsänderung $\Delta_{ij}(t)$ mit dem Faktor η^- reduziert. Ohne Vorzeichenwechsel beschleunigt η^+ die Konvergenz in flachen Bereichen der Gesamtfehlerfunktion. Die Verbindungsgewichte im Iterationszeitpunkt $(t+1)$ ergeben sich zu

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) . \quad (92)$$

Nach Untersuchungen durch RIEDMILLER sind die Wichtungskonstanten mit den Werten $\eta^+ = 1.2$ und $\eta^- = 0.5$ optimal belegt.

Rprop ist ein *offline*-Trainingsverfahren, d.h. zu jedem Iterationsschritt erfolgt die Anpassung der Verbindungsgewichte erst nachdem alle Trainingsmuster dem Netz präsentiert sind. Dies ermöglicht die Realisation von Rprop in SNNS mit einer bzgl. Gleichung (85) erweiterten Gesamtfehlerfunktion

$$E_{\text{erw}} = \text{SSE} + 10^{-\alpha} \sum w_{ij}^2 , \quad (93)$$

die betragsmäßig große Gewichte bestraft. Die drei Steuerparameter des Lernverfahrens im Simulator SNNS sind Δ_0 als Startwert des Betrags der Gewichtsänderung, Δ_{max} als maximal zulässiger Änderungswert und α als Parameter zur betragsmäßigen Verkleinerung der Verbindungsgewichte.

5.5 Pruning

Ziel jedes maschinellen Lernverfahrens ist es, den gesuchten, den Lerndaten inhärenten funktionalen Zusammenhang mit minimaler Systemkomplexität bei gleichzeitig maximaler Generalisierungsfähigkeit möglichst genau zu bestimmen. Die Vereinfachung von Neuronalen Netzen mit Hilfe von Ausdünnungsverfahren (*Pruning*-Verfahren) auf eine für die jeweilige Problemstellung minimal notwendige Netztopologie ist dort sinnvoll, wo die Festlegung einer problemnahen Netztopologie a priori nicht möglich ist (siehe z.B. Kapitel 8).

Grundidee ist die Reduzierung allgemeiner Ansatztopologien auf zugehörig zur jeweiligen Problemstellung minimal notwendige Netztopologien. Als Ausgangsnetzwerke werden sowohl vollständig vernetzte konventionelle Netze (siehe Kapitel 8) als auch Netze mit eingebautem Funktionsansatz (siehe Abschnitt 6.4) verwendet.

Verfahren zur Reduzierung von *Feedforward*-Netztopologien gehen im allgemeinen von überdimensionierten konventionellen Netzen aus, die die gesuchte und zur aktuellen Problemstellung minimal notwendige Netzübertragungsfunktion potentiell beinhalten. Gleiches gilt für Ausgangsnetztopologien mit eingebautem Funktionsansatz, der den gesuchten und in den Trainingsdaten vermuteten funktionalen Zusammenhang umfassen muß.

Unter Reduzierung ist je nach verwendetem Algorithmus sowohl das Löschen von Gewichten mit der zugehörigen Verbindung als auch das Löschen von Neuronen selbst zu verstehen. Vorteile, die sich aus der Reduzierung der Systemkomplexität durch Vereinfachung der Netztopologie ergeben, sind:

- Mögliche Verbesserung der Generalisierungsleistung des trainierten Netzes. Dies ist für konventionelle Netze umstritten.
- Die Vereinfachung einer allgemeinen Netzansatzfunktion kann zu einem besseren Verständnis der zugrundeliegenden physikalischen Problemstellung führen.
- Bei eingebautem Funktionsansatz kann eine optimal reduzierte und vollständig trainierte Netztopologie die maximal mögliche Generalisierungsleistung innerhalb des Definitionsbereichs des beschriebenen Zusammenhangs erreichen. Der in den Trainingsdaten inhärent gespeicherte funktionale Zusammenhang ist damit analytisch exakt erfaßt.
- Schnellere Antwortzeiten reduzierter Netze sowie Verringerung des Trainings-, Speicher- und Hardwareaufwands

Die reduzierte Komplexität der vereinfachten Netztopologie kommt dann auch einer möglichen VLSI-Realisierung des Neuronalen Netzes zugute.

5.5.1 Ausdünnungsverfahren

Pruning-Verfahren zur Reduzierung Neuronaler Netze lassen sich im wesentlichen in drei Kategorien einteilen (siehe REED [115]):

- Die in der Arbeit eingesetzten Sensitivitätsverfahren untersuchen den Einfluß jedes Gewichts auf die Gesamtfehlerfunktion und löschen iterativ bis zu einer definierten Grenze diejenigen Gewichte mit zugehöriger Verbindung, die den geringsten Einfluß auf den Trainingsfehler haben.
- Straftermverfahren erweitern die Gesamtfehlerfunktion E um einen Kostenterm, der monoton mit der Größe der freien Parameter im Netz steigt. Als Resultat verändert der Lernalgorithmus Gewichte, die während des Trainings nicht verstärkt werden, stetig gegen null. Dem Ansatz liegt die Annahme zugrunde, daß durch Lernverfahren nicht beeinflusste Gewichte auch keinen Einfluß auf das untersuchte Problem haben (siehe z.B. SCHREINER [116]).

- Anpassung einer Netztopologie mit stochastischen Optimierungsverfahren, z.B. Genetische Algorithmen (siehe z.B. MORIARTY UND MIKKULAINEN [117], RUDOLPH [17, 15], WHITLEY UND BOGART [118] sowie GRÜNINGER [119]).

Das Löschen eines Verbindungsgewichts bedingt gleichzeitig immer auch das Löschen der zugehörigen gerichteten Verbindung. Verdeckte Neuronen werden nur dann gelöscht, wenn diese nicht mehr auf einem Pfad von der Eingabe- zur Ausgabeschicht liegen. Dies ist der Fall, wenn z.B. noch eine Informationsverbindung zur Eingabeschicht, aber keine Informationsverbindung zur Ausgabeschicht mehr besteht. Das Löschen von Neuronen wirkt auch rekursiv, d.h. ganze Verbindungsstrukturen können gelöscht werden (siehe Abbildung 8).

Eingabe- und Ausgabeneuronen werden auch im isolierten Zustand nicht gelöscht, da sie durch den Ansatz der Netzübertragungsfunktion sowie durch das Format der Trainingsdaten festgelegt sind. Aus der Isolation eines Ausgabeneurons kann aber auf dessen Bedeutung in der Netzübertragungsfunktion und damit im gesuchten Zusammenhang geschlossen werden (siehe Abschnitt 6.4).

Bei allen Methoden ist es schwierig, festzulegen, wann alle überflüssigen Verbindungen gelöscht sind. Der Ausdünnungsvorgang muß gestoppt werden, sobald jede weitere Veränderung die Netztopologie bzw. die zugehörige mathematische Netzübertragungsfunktion so einschränkt, daß eine Erfassung der untersuchten Problemstellung nicht mehr möglich ist.

Als Beispiel sei ein Polynomansatz der Form

$$f(x_1, x_2) = w_{1\ 6} x_1 + w_{4\ 6} x_2 + w_{5\ 6} x_1^{w_2\ 5} x_2^{w_3\ 5} \quad (94)$$

genannt. Er entspricht der Übertragungsfunktion des Neuronalen Netzes in Abbildung 8 (links). Das Löschen des Gewichts $w_{5\ 6}$ im Netzwerk (Abbildung 8 (links)) bewirkt das Löschen aller Informationsverbindungen des Neurons 5 zur Ausgabeschicht und als Folge das Löschen des Neurons 5 selbst (siehe Abbildung 8 (rechts)). Dies reduziert den Polynomansatz aus Gleichung (94) zu

$$f(x_1, x_2) = w_{1\ 6} x_1 + w_{4\ 6} x_2 . \quad (95)$$

Als Folge kann dann kein gemischtes Glied $x_1 x_2$ mehr berücksichtigt werden und die Ansatzfunktion erfaßt den inhärenten Zusammenhang der Trainingsdaten nicht oder nur lokal. Dieser Fehler ist auch durch beliebig langes Nachtrainieren nicht mehr zu beheben. Insbesondere beim *Pruning* problemspezifischer Neuronaler Netze kann dies zur Funktionsuntüchtigkeit des ausgedünnten Netzes führen.

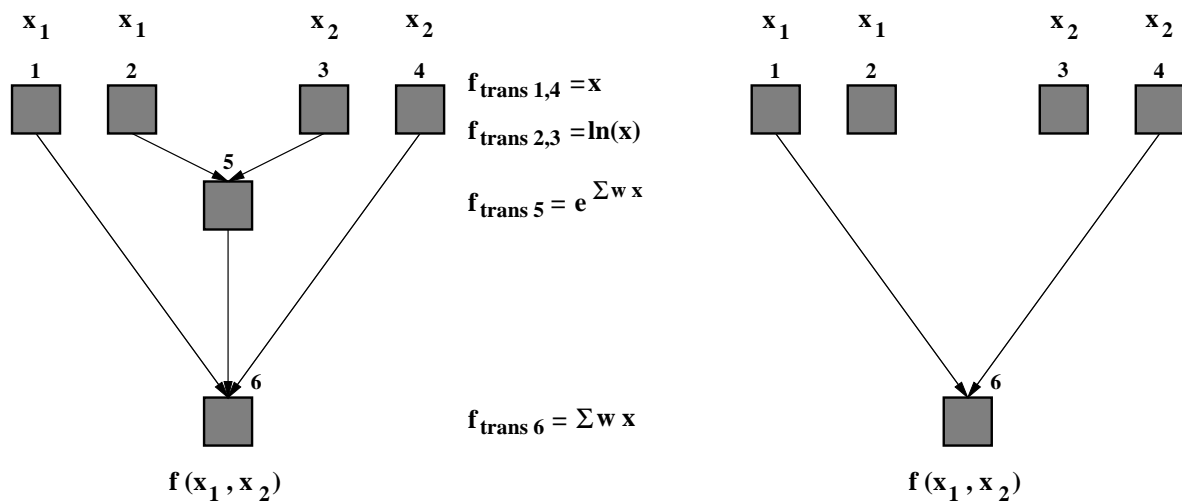


Abbildung 8: Auswirkungen von rekursivem Löschen

Sensitivitätsverfahren erkennen keine logische Korrelationen von Neuronenteilstrukturen. Als Folge können auch korrelierte Teilstrukturen durch Löschen einer einzigen Verbindung entfernt werden. Damit ist klar, daß nicht alle *Pruning*-Läufe erfolgreich sein können. Grund ist die zufällige Initialisierung der Gewichtsmatrix vor Trainingsbeginn.

5.5.2 Optimal Brain Surgeon

Im folgenden wird das in der vorliegenden Arbeit eingesetzte *Pruning*-Verfahren *Optimal Brain Surgeon* (OBS) nach HASSIBI UND STORK [120] sowie HASSIBI ET AL. [121, 122] vorgestellt. Die Auswahl basiert auf dem Studium zugehöriger Literatur sowie eigenen Vergleichsrechnungen mit *Optimal Brain Damage* (OBD) nach LE CUN ET AL. [123] und *Magnitude Based Pruning* (MBP) nach HERTZ ET AL. [124]. Detaillierte Beschreibungen zu MBP, OBD sowie zu weiteren *Pruning*-Verfahren wie z.B. *Skeletonization* finden sich z.B. bei REED [115] und SCHREINER [116].

OBS als Sensitivitätsverfahren nutzt Informationen der zweiten Ableitungen der Gesamtfehlerfunktion nach den Netzgewichten zur Bestimmung derjenigen Netzgewichte, deren Löschen mit der geringsten Zunahme des Gesamtfehlers verbunden ist. Gleichzeitig werden die jeweils übrig gebliebenen Gewichte als Resultat der gelösten, restringierten Minimierungsaufgabe aus Gleichung (99) an die veränderte Situation angepaßt.

Die Gesamtfehlerfunktion eines Neuronalen Netzes mit m Ausgabeneinheiten und p präsentierten Trainingsmustern ist durch Gleichung (85) gegeben. Der Anstieg der Gesamtfehlerfunktion E als Resultat auf das Löschen von Gewichten wird durch eine Taylorreihenentwicklung

der Gesamtfehlerfunktion in der Form

$$\Delta E = \left(\frac{\partial E}{\partial \mathbf{w}} \right)^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} + \mathcal{O} (\|\Delta \mathbf{w}\|^3) \quad (96)$$

angenähert. Die Gewichtsmatrix \mathbf{W} sei zur Vereinfachung als n_w -dimensionaler Gewichtsvektor \mathbf{w} angenommen und

$$\mathbf{H} = \nabla_{\mathbf{w}\mathbf{w}}^2 E = \frac{\partial^2}{\partial^2 \mathbf{w}} E \quad (97)$$

bezeichnet die Hesse-Matrix der Gesamtfehlerfunktion E , während \mathcal{O} alle Terme dritter und höherer Ordnung der Reihenentwicklung umfaßt. Folgende Annahmen vereinfachen die Näherung:

- Der aktuelle Trainingszustand des Netzwerks habe ein lokales Minimum der Gesamtfehlerfunktion erreicht.
- Die Umgebung des Minimums sei lokal quadratisch.

Damit verschwinden aus Gleichung (96) der lineare Term und alle Terme dritter und höherer Ordnung. Das Löschen des Gewichts w_q ist durch

$$\mathbf{e}_q^T \Delta \mathbf{w} + w_q = 0 \quad (98)$$

definiert, wobei $\Delta \mathbf{w}_q = \mathbf{e}_q^T \Delta \mathbf{w}$ die Änderung des Gewichts q in Richtung des Einheitsvektors \mathbf{e}_q in dem durch \mathbf{w} aufgespannten Gewichtsraum um den Wert $\Delta \mathbf{w}$ darstellt. Damit ergibt sich die restringierte Optimierungsaufgabe

$$\min_q \left[\min_{\Delta \mathbf{w}} \left(\frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} \right) \mid \mathbf{e}_q^T \Delta \mathbf{w} + w_q = 0 \right], \quad (99)$$

die durch Bildung der funktionalen Ableitungen der zugehörigen Lagrange-Gleichung

$$L = \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} + \lambda (\mathbf{e}_q^T \Delta \mathbf{w} + w_q) \quad (100)$$

nach $\Delta \mathbf{w}$ und λ (Restriktionen) gelöst werden kann. Man erhält die optimale Änderung aller restlichen Gewichte für den Fall, daß das Gewicht w_q gelöscht wird, zu

$$\Delta \mathbf{w} = - \frac{w_q}{[\mathbf{H}^{-1}]_{qq}} \mathbf{H}^{-1} \mathbf{e}_q, \quad (101)$$

wobei für $[\mathbf{H}^{-1}]_{qq} = \mathbf{e}_q^T \mathbf{H}^{-1} \mathbf{e}_q$ gilt. Die Erhöhung des Gesamtfehlers (*saliency*) aus dem Löschen eines Gewichts w_q folgt zu

$$L_q = \frac{1}{2} \frac{w_q^2}{[\mathbf{H}^{-1}]_{qq}}. \quad (102)$$

Durch Gleichung (101) werden nach Löschen des Gewichts w_q alle übrigen Gewichte des Netzwerks an die veränderte Situation angepaßt. Der Algorithmus kann damit nach HASSIBI UND STORK [120] sowie HASSIBI ET AL. [121, 122] wie folgt angegeben werden:

1. Auswahl einer problemgerechten Netzwerkarchitektur
2. Training des Netzwerks in ein (lokales) Minimum der Gesamtfehlerfunktion
3. Berechnung von \mathbf{H}^{-1} für alle Gewichte q des Netzwerks
4. Bestimmung der *saliency* L_q des Netzwerks gemäß Gleichung (102) für alle w_q
5. Falls der kleinste *saliency*-Wert eines Gewichts w_q sehr viel kleiner als E ist, Gewicht q löschen. Gehe zu Schritt 6, sonst zu Schritt 7
6. Anpassung aller restlichen Gewichte gemäß Gleichung (101). Nachtraining um festgelegte Anzahl von Epochen. Zurück zu Schritt 3
7. Weiteres Löschen von Gewichten führt zu starkem Anwachsen von E . Den Zustand vor dem Löschen des letzten Gewichts wieder herstellen (fakultativ).

Die für jedes Löschen eines Netzgewichts notwendige Berechnung der inversen Hesse-Matrix \mathbf{H}^{-1} wird durch einen von HASSIBI UND STORK in [120] gegebenen Algorithmus in einem einzigen Durchgang durch alle Trainingsmuster $i = 1, \dots, p$ ermöglicht. Eine detaillierte Darstellung hierzu findet sich bei SCHREINER [116].

5.5.3 Diskussion des OBS-Algorithmus

Grundproblem des besprochenen Sensitivitätsverfahrens OBS ist der hohe Rechenaufwand bei der Ermittlung des *saliency*-Werts zur Auswahl des zu löschenden Netzgewichts. OBS arbeitet bei den Berechnungen der inversen Hesse-Matrix ohne einschränkende Vereinfachungen. Dies hat folgende Konsequenzen:

- Hoher CPU-Aufwand: Zu jedem *Pruning*-Schritt ist die Berechnung einer inversen Hesse-Matrix notwendig. Die Zeitkomplexität pro Matrixinversion liegt in der Ordnung $O(m p n_w^2)$.
- Hoher Speicherbedarf: Der Speicherbedarf für eine Hesse-Matrix liegt in der Ordnung $O(n_w^2)$.

Die Variable m bezeichnet die Anzahl der Ausgabeneuronen, p die Anzahl der Trainingsmuster und n_w die Anzahl der Gewichte im Netzwerk. Die genannten Eigenschaften von OBS machen das Verfahren für sehr große Anwendungen (z.B. 10000 Gewichte) ungeeignet. Gleichzeitig zeigte sich aber auch, daß OBS für die Reduktion allgemeiner Netztopologien mit eingebautem Funktionsansatz, wenn es besonders darauf ankommt, die exakt richtigen (überflüssigen) Verbindungen zu löschen, das geeignetste Verfahren ist. Als Beispiel sei die Ausdünnung allgemeiner Netztopologien mit eingebautem Funktionsansatz genannt. Da die Ausdünnungsergebnisse z.T. sehr sensitiv gegenüber Parameteränderungen sind, empfiehlt es sich stets, mehrere Simulationsrechnungen mit unterschiedlichen Parametereinstellungen durchzuführen (siehe Abschnitt 6.4).

5.5.4 Realisation von OBS in SNNS

OBS ist aus seiner Konzeption heraus nicht in der Lage, ein eigenes Kriterium zur Beendigung des Algorithmus zu ermitteln. Dies muß durch eine externe Steuerung realisiert werden. Zusätzlich benötigt das Gesamtverfahren einen dem *Pruning*-Algorithmus untergeordneten Lernalgorithmus für Vor-, Zwischen- und Nachtraining des untersuchten Netzwerks. Die Implementierung von OBS im Netzsimulator SNNS durch SCHREINER [116] arbeitet mit fünf Steuerungsparametern:

1. Anzahl der Lernzyklen vor Beginn des *Pruning*-Algorithmus *Maximum Learning Cycles* (MLC) sowie für das Training zwischen bzw. nach den einzelnen *Pruning*-Schritten *Maximum Retraining Cycles* (MRC)
2. Parameter des untergeordneten Lernalgorithmus (BackpropMomentum oder Rprop)
3. *Minimum Error to Stop* (MES): Absoluter Netzfehler, dessen Unterschreiten das Training beendet; soll Übertrainieren verhindern.
4. *Maximum Error Increase* (MEI): Die zulässige obere Fehlerschranke ergibt sich als das $(1 + \frac{MEI}{100})$ -fache des Netzfehlers nach dem Training.
5. *Accepted Error* (AE): Absoluter Netzfehler, der zur Beendigung des Algorithmus überschritten werden muß.

Abbruchkriterium für den Ausdünnungsalgorithmus ist das Überschreiten beider Parameter 4 und 5 durch den Netzfehler des Nachtrainings nach einem *Pruning*-Schritt. Problem bei allen

Pruning-Verfahren ist die Wahl der richtigen Steuerungsparameter, da schon kleine Unterschiede z.B. bei MEI und AE und in noch stärkerem Maße bei den Parametern des untergeordneten Lernverfahrens zu sehr unterschiedlichen Ausdünnungsergebnissen führen können.

5.6 Leistungsfähigkeit von Feedforward-Netzwerken

Die Leistungsfähigkeit eines *Feedforward*-Netzwerks läßt sich u.a. anhand seiner Repräsentationsfähigkeit, Lernfähigkeit und Komplexität sowie anhand seiner Generalisierungsleistung beurteilen.

5.6.1 Repräsentationsfähigkeit, Lernfähigkeit und Komplexität

Repräsentationsfähigkeit stellt ein Potential des Netzwerks dar, einen über die Trainingsdaten gegebenen funktionalen Zusammenhang durch optimale Anpassung der variablen Parameter des Netzwerks (Verbindungsgewichte, Netzplan, Neuronenfunktionalität) in der Netzübertragungsfunktion zu erfassen.

Lernfähigkeit ist eine Eigenschaft des gewählten Lernverfahrens und beschreibt dessen Fähigkeit, die variablen Parameter des Netzwerks so zu bestimmen, daß zusammen mit der Repräsentationsfähigkeit der in den Trainingsdaten repräsentierte Zusammenhang exakt erfaßt werden kann. Repräsentations- und Lernfähigkeit sind damit sich gegenseitig begrenzende Faktoren, die nur optimal kombiniert die maximale Leistungsfähigkeit des Netzwerks ermöglichen.

Mit Hilfe der *O*-Notation läßt sich die asymptotische obere Schranke einer Komplexitätsklassifikation angeben. Die Zeitkomplexität bei der Berechnung der inversen Hesse-Matrix \mathbf{H}^{-1} im *Pruning*-Verfahren OBS ist $O(m p n_w^2)$. Das Training einer festen *Feedforward*-Netztopologie (kein *Pruning*) für E_p Epochen mit dem Lernalgorithmus BackpropMomentum hat dagegen die Zeitkomplexität $O(E_p p n_w)$. Reines Netztraining benötigt damit um Größenordnungen geringere Rechenressourcen als das Ausdünnungsverfahren OBS. Theoretische Untersuchungen finden sich z.B. bei SCHÖNING [125] und ROJAS [102].

5.6.2 Generalisierungsleistung

Die Generalisierungsleistung ist als Fähigkeit eines trainierten Neuronalen Netzes definiert, auf unbekannte Testdaten die richtige oder im Vergleich zur Sollausgabe eine möglichst minimal davon abweichende Istaussgabe zu liefern. Die in Abschnitt 5.4.1 in den Gleichungen (86)

und (87) gegebenen Kenngrößen erlauben eine statistische Auswertung der Generalisierungsleistung der untersuchten Netzwerke bezüglich der verwendeten Testdaten.

Ist die Generalisierungsfähigkeit eines trainierten oder durch *Pruning* reduzierten konventionellen Netzwerks gering, so arbeitet das Netzwerk u.U. als sogenannte Zuordnungstabelle ohne ausreichende Interpolation der zu beschreibenden Funktion zwischen den Stützstellen der Trainingsmuster. Eine solche Zuordnungstabelle liefert für die bekannten Lernmuster zwar geringe Fehlerwerte, zeigt aber für unbekannte Testdaten, auch wenn diese aus dem Intervall der Trainingsdaten stammen, hohe Abweichungen zwischen Soll- und Istausgabe.

Ein problemspezifisches Neuronales Netz besitzt dann die maximale Generalisierungsfähigkeit, wenn durch die Netzübertragungsfunktion des erfolgreich trainierten Netzwerks der in den Trainingsdaten beschriebene funktionale Zusammenhang exakt erfaßt wird. Das Netz ist dann in der Lage, über dem gesamten und damit nicht auf das Parameterintervall der Trainingsdaten beschränkten Definitionsbereich des erfaßten Zusammenhangs korrekte Aussagen auf unbekannte Testdaten zu liefern. Dies wird zusammen mit der notwendigen Datenvorverarbeitung durch Dimensionsanalyse in Abschnitt 6.1.1 detaillierter untersucht.

Eine optimale Wahl der Lern- und *Pruning*-Parameter für Training und *Pruning* Neuronaler Netze ist vor Trainingsbeginn nicht möglich. Zur Wahl der Lern- und *Pruning*-Parameter gibt es im wesentlichen zwei Alternativen: Die Untersuchung verschiedener Parametervariationen (z.B. durch ein Optimierungsverfahren) und Erfahrungswerte des Anwenders. Bei den in den folgenden Anwendungskapiteln dargestellten Ergebnissen handelt es sich deshalb immer um erfolgreiche Trainings- und *Pruning*-Resultate der zahlreichen untersuchten Parametervariationen basierend auf den Erfahrungen des Autors.

Das folgende Kapitel erläutert die Erfassung ausgewählter Systemantworten von Tragwerkstrukturen durch geeignete Neuronale Netze. Insbesondere die Beschreibung von Optimierungsrestriktionen (Knotenverschiebungen, Stabspannungen) ist Ziel dieser Untersuchungen.

6 Neuronale Netze in der Strukturmechanik

Die zur Tragwerksoptimierung eingesetzten Algorithmen benötigen zu jedem Iterationsschritt aktuelle Systemantworten (Zielfunktion, Restriktionen) der untersuchten Struktur. Diese wurden bisher mit Hilfe der FE-Methode bereitgestellt. Die hier vorgeschlagene Methode sieht die Ermittlung der zur Tragwerksoptimierung notwendigen Systemantworten durch trainierte Neuronale Netze vor. Für die zahlreichen notwendigen Bestimmungen von Zielfunktion und Restriktionen zu den aktuellen Parametern im Iterationspunkt des Optimierungsverfahrens bietet die Vorgehensweise aufgrund der kurzen Antwortzeiten trainierter Neuronaler Netze signifikante Vorteile. Dies gilt insbesondere für stochastische Optimierungsverfahren wie die in Kapitel 7 eingesetzten Evolutionsstrategien.

Zur Beschreibung physikalischer Vorgänge, zu denen keine Kenntnisse des zu erwartenden Zusammenhangs vorliegen, bietet sich der Einsatz konventioneller Neuronaler Netze an. Dies bedeutet aber eine Einschränkung des Generalisierungspotentials der trainierten, konventionellen Netztopologie auf den Wertebereich der verwendeten Trainingsdaten und wird deshalb für die Optimierungsuntersuchungen nur zur Erfassung von Strukturzuverlässigkeiten als ein Kriterium der Mehrzieloptimierungsaufgabe eingesetzt (siehe Abschnitt 7.2).

Das vorgeschlagene Grundkonzept sieht die Aufstellung allgemeiner Ansatzfunktionen und ihre identische Abbildung in Neuronale Netze nach der in Abschnitt 5.2.3 beschriebenen Methodik vor. Die Ermittlung der Variablen der Ansatzfunktion zur angestrebten analytisch exakten Erfassung von Systemantworten (Knotenverschiebungen, Stabspannungen) eines statisch belasteten Fachwerks erfolgt auf zwei verschiedene Vorgehensweisen:

- Training der Variablen der Ansatzfunktion (Verbindungsgewichte). Überflüssige Variablen werden durch Training zugehöriger Verbindungsgewichte auf null eliminiert.
- Eliminierung überflüssiger Variablen der Ansatzfunktion durch *Pruning* der Ausgangsnetztopologie und anschließendes Nachtraining der Restvariablen der reduzierten Netztopologie.

Die im folgenden angegebenen Resultate stellen erfolgreiche Simulationsergebnisse dar.

6.1 Beschreibung von Systemantworten mit problemspezifischen Neuronalen Netzen

In den folgenden Abschnitten werden Möglichkeiten untersucht, durch problemspezifische Neuronale Netze die Systemantworten Knotenverschiebungen und Stabspannungen statisch belasteter Fachwerkstrukturen zu erfassen. Ziel ist die Generierung trainierter Neuronaler Netze, die Systemreaktionen exakt und mit minimalen Antwortzeiten erfassen und einem Algorithmus z.B. im Rahmen von Optimierungsuntersuchungen zur Verfügung stellen.

6.1.1 Dimensionsanalyse

Zur Erfassung der Systemreaktionen Knotenverschiebungen und Stabspannungen durch problemspezifische Neuronale Netze erfolgt die Datenvorverarbeitung der Test- und Trainingsdaten durch Dimensionsanalyse. Die im folgenden dargestellten Beispiele statisch belasteter, ebener Fachwerke verdeutlichen die kombinierte Methodik aus Dimensionsanalyse und Abbildung dimensionsloser Zusammenhänge in problemspezifische Neuronale Netze.

Die Dimensionsanalyse ermöglicht durch das in Abschnitt 2.3 beschriebene Π -Theorem die äquivalente Umwandlung der dimensionshomogenen Beziehung f eines untersuchten physikalischen Vorgangs in einen Zusammenhang G dimensionsloser Π -Größen. Die Approximation von f durch trainierte Neuronale Netze verlagert sich damit aus dem n -dimensionalen Ortsraum dimensionsbehaltender Parameter (x_1, x_2, \dots, x_n) in den p -dimensionalen Bildraum dimensionsloser Π -Größen des äquivalenten Zusammenhangs G (siehe Gleichung (26)). Die Anzahl n dimensionsbehaltender Parameter, die zur vollständigen Erfassung der dimensionshomogenen Problemstellung f notwendig sind, reduziert sich durch Dimensionsanalyse um den Rang r der Dimensionsmatrix \mathbf{A} . Der untersuchte physikalische Sachverhalt ist damit ohne Informationsverlust äquivalent durch $p = n - r$ dimensionslose Π -Größen beschrieben. Ein geeignetes, problemspezifisches und mit dimensionslosen Π -Größen erfolgreich trainiertes Neuronales Netz ist in der Lage, die Funktion G exakt zu erfassen.

Jede dimensionslose Größe Π_i umfaßt mit einem Wert theoretisch unendlich viele Realisationen aus den Variationsmöglichkeiten ihrer dimensionsbehafteten Ausgangsparameter \mathbf{x} zugehörig zum Exponentenvektor \mathbf{k}_i und zeigt so das ihr inhärente Generalisierungspotential. Die Rückumwandlung dimensionsloser Π -Größen in ihre dimensionsbehafteten Ausgangsparameter \mathbf{x} ist durch die Transformationsvorschrift $\phi_i(\mathbf{x}, \mathbf{k}_i)$ nach Gleichung (21) möglich. Voraussetzung hierzu ist, daß alle dimensionsbehafteten Parameter außer der gesuchten Größe bekannt sind.

Zur Dimensionsanalyse ist die Relevanzliste der untersuchten Problemstellung zu erstellen. Die Relevanzliste erfaßt in einer Dimensionsmatrix \mathbf{A} alle für eine vollständige Beschreibung des untersuchten physikalischen Vorgangs relevanten dimensionsbehafteten Größen sowie gesuchte Zielgrößen. Die Auswertung von \mathbf{A} liefert über die Exponentenvektoren \mathbf{k}_i der Maßzahlen (x_1, x_2, \dots, x_n) die Transferfunktionen ϕ_i der p dimensionslosen Potenzprodukte Π_i in der Form

$$\Pi_i = \phi_i(\mathbf{x}, \mathbf{k}_i) = x_1^{k_{i1}} x_2^{k_{i2}} \cdot \dots \cdot x_n^{k_{in}} \quad \text{mit } i = 1, \dots, p. \quad (103)$$

Die Wahl des Exponentenvektors $\mathbf{k}_i = (k_{i1}, k_{i2}, \dots, k_{in})$ zum dimensionslosen Potenzprodukt Π_i für $i = 1, \dots, p$ ist frei, solange die \mathbf{k}_i ein Fundamentalsystem von Lösungsvektoren des linearen Gleichungssystem $\mathbf{K} \mathbf{A} = \mathbf{0}$ bilden.

Ist die Funktion G nach erfolgreichem Training der zugehörigen problemspezifischen Netztopologie exakt durch deren Netzübertragungsfunktion abgebildet, so liefert das trainierte Neuronale Netz im Gültigkeitsbereich des zugrundeliegenden physikalischen Modells auf beliebige dimensionslose Eingabegrößen korrekte dimensionslose Ausgaben. Damit ist dann gleichzeitig auch der physikalische Vorgang f im Ortsraum für beliebige Ausgangsparameter \mathbf{x} innerhalb des Gültigkeitsbereichs des zugrundeliegenden physikalischen Modells eindeutig erfaßt. Dies werde als maximale Generalisierungsfähigkeit eines Neuronalen Netzes bezeichnet.

6.1.2 Charakteristische Kenngrößen winkeltreuer ebener Fachwerkstrukturen

Die Relevanzliste des Fachwerks FW1 in Abbildung 9 umfaßt die äußere Last P , die Elastizitäten $S_{j,k} = E_{j,k} A_{j,k}$ der kreisförmigen Stabquerschnitte zwischen den Knoten j und k mit Elastizitätsmodul $E_{j,k}$ und Stabquerschnittsfläche $A_{j,k}$, die charakteristischen Stablängen a und b sowie die gesuchte Verschiebung des Knotens 2 in y -Richtung $\delta_{K2,y}$. Die Nomenklatur gilt sinngemäß für alle folgenden Fachwerke.

Die Methodik der Erfassung geometrischer Charakteristika in einer dimensionslosen Kenngröße ist in der vorgestellten Form auf ebene, rechtwinklige Fachwerkstopologien, wie in Abbildung 9 dargestellt, beschränkt. Für die Fachwerke sollen bzgl. geometrischer Variationen die Voraussetzungen winkeltreuer Abbildungen im Sinne von NAAS UND SCHMID [126] gelten. Die im folgenden getroffenen Aussagen sind beschränkt auf statisch bestimmte Fachwerke unter den Annahmen der Theorie I. Ordnung. Es gelten die Annahmen, daß die Topologie des Fachwerks nicht verändert werde und die Verformungen des Fachwerks infolge der konstanten, statischen Last klein gegenüber den Abmessungen des Fachwerks seien (siehe auch Abschnitt 3.1.1).

Die geometrische Kenngröße $\Psi = \frac{a}{b}$ beschreibt das Verhältnis der charakteristischen Stablängen des Fachwerks und sei innerhalb eines Fachwerks konstant. Der für ein Fachwerk jeweils konstante Wert $\Psi_i = \text{konst}_i$ umfaßt beliebig viele Kombinationsmöglichkeiten der Werte a zu b . Die über Training bzw. *Pruning* geeigneter problemspezifischer Neuronaler Netze ermittelten Netzübertragungsfunktionen sind in der Lage, in den Trainingsdaten berücksichtigte dimensionslose Systemantworten (z.B. Knotenverschiebungen wie $\frac{\delta_{Kn\ 2,y}}{a}$) für Fachwerke mit beliebigen Längenverhältnissen Ψ analytisch exakt zu beschreiben. Mit der Vorgehensweise können physikalische Vorgänge über dem gesamten Gültigkeitsbereich des zugrundeliegenden physikalischen Modells vollständig erfaßt werden.

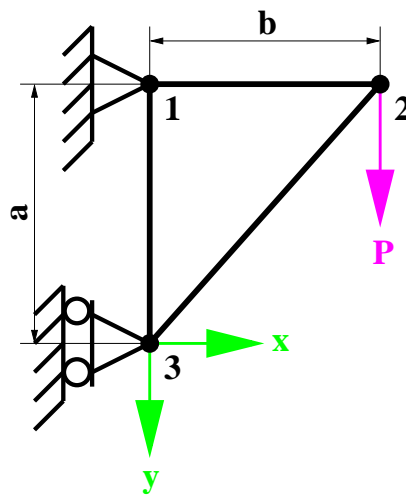


Abbildung 9: FW1: Statisch bestimmtes Fachwerk mit drei Stäben

6.1.3 Datenvorverarbeitung

Es werden zwei Methoden zur Vorverarbeitung von Test- und Trainingsdaten Neuronaler Netze besprochen: Dimensionsanalyse und konventionelle Transformation. Anwendungsbeispiel beider Methoden ist das Fachwerk FW1 aus Abbildung 9.

Das mechanische Grundgrößensystem (M, L, T) setzt sich aus den primären Grundgrößen Masse M , Länge L und Zeit T zusammen. Die Maßzahl der äußeren Last P hat genauso wie die Maßzahl der Elastizität $S_{j\ k}$ die Dimension $[P] = [S_{j\ k}] = M L T^{-2}$, während die Maßzahlen der charakteristischen Längen a und b sowie der Auslenkung $\delta_{Kn\ 2,y}$ die Dimension $[a] = [b] = [\delta_{Kn\ 2,y}] = L$ besitzen.

Für das Fachwerk FW1 ergibt sich zur Relevanzliste aus Abschnitt 6.1.2 die in Abbildung 10 links dargestellte Dimensionsmatrix \mathbf{A}_1 . Der Rang von \mathbf{A}_1 ist zwei; damit genügen

\mathbf{A}_1	M	L	T
P	1	1	-2
$S_{1\ 2}$	1	1	-2
$S_{2\ 3}$	1	1	-2
$S_{1\ 3}$	1	1	-2
a	0	1	0
b	0	1	0
$\delta_{Kn\ 2,y}$	0	1	0

$$\mathbf{K} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

Abbildung 10: Dimensionsmatrix \mathbf{A}_1 und Exponentenmatrix \mathbf{K} zu Fachwerk FW1

$p = 5$ dimensionslose Potenzprodukte zur vollständigen Erfassung der Problemstellung. Eine mögliche Exponentenmatrix \mathbf{K} der fünf Potenzprodukte ist in Abbildung 10 rechts dargestellt. Sie erfüllt zusammen mit der Dimensionsmatrix \mathbf{A}_1 die notwendige Bedingung des homogenen linearen Gleichungssystems in Gleichung (23) bzw. (24). Aus der Anwendung der Dimensionsanalyse ergeben sich mit der Exponentenmatrix \mathbf{K} gemäß der Transformationsvorschrift ϕ_i in Gleichung (103) fünf dimensionslose Potenzprodukte der Form

$$\Pi_1 = \frac{P}{S_{1\ 2}} \quad \Pi_2 = \frac{P}{S_{2\ 3}} \quad \Pi_3 = \frac{P}{S_{1\ 3}} \quad \Psi = \frac{a}{b} \quad \Delta_{2,y} = \frac{\delta_{Kn\ 2,y}}{a}. \quad (104)$$

In Fachwerk FW1 hat die Auslenkung des Knotens 2 in y-Richtung nach dem Einheitslastgesetz die Form

$$\delta_{Kn\ 2,y} = \frac{P}{S_{12}} \frac{a}{\Psi^3} + \frac{P}{S_{23}} a \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}} + \frac{P}{S_{13}} a. \quad (105)$$

Mit den Transformationsvorschriften aus Gleichung (104) folgt in dimensionsloser Form

$$\Delta_{2,y} = \Pi_1 \frac{1}{\Psi^3} + \Pi_2 \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}} + \Pi_3. \quad (106)$$

In Abschnitt 6.2 und 6.4 wird die Aufstellung verschiedener problemspezifischer Netztopologien besprochen, die das Potential besitzen, die dargestellten Zusammenhänge analytisch exakt zu erfassen.

Eine nicht auf Dimensionsanalyse basierende dimensionslose Darstellung auftretender Normalspannungen der Stäbe des Fachwerks FW1 nimmt durch die Transformationsvorschriften

$$\begin{aligned} \Theta_1 &= \frac{P}{A_n \sigma_n} & \Theta_2 &= \frac{A_{1\ 2}}{A_n} & \Theta_3 &= \frac{A_{2\ 3}}{A_n} & \Theta_4 &= \frac{A_{1\ 3}}{A_n} \\ \Gamma_1 &= \frac{\sigma_{1\ 2}}{\sigma_n} & \Gamma_2 &= \frac{\sigma_{2\ 3}}{\sigma_n} & \Gamma_3 &= \frac{\sigma_{1\ 3}}{\sigma_n} & \Psi &= \frac{a}{b} \end{aligned} \quad (107)$$

die einfach in ein Neuronales Netz zu implementierende Form

$$\begin{aligned} \sigma_{23} &= -\frac{P}{A_{23}} \left(1 + \frac{1}{\Psi^2}\right)^{\frac{1}{2}} \Leftrightarrow \Gamma_2 = -\frac{\Theta_1}{\Theta_3} \left(1 + \frac{1}{\Psi^2}\right)^{\frac{1}{2}} \\ \sigma_{12} &= \frac{P}{A_{12}} \Psi^{-1} \Leftrightarrow \Gamma_1 = \frac{\Theta_1}{\Theta_2} \Psi^{-1} \quad \sigma_{13} = \frac{P}{A_{13}} \Leftrightarrow \Gamma_3 = \frac{\Theta_1}{\Theta_4} \end{aligned} \quad (108)$$

an. Die Variable A_{jk} beschreibt die Stabquerschnittsfläche des kreisförmigen Stabs $Stab_{jk}$ zwischen den Knoten j und k mit der Normalspannung σ_{jk} . Die Normierungsgrößen A_n und σ_n dienen der Erzeugung numerisch ausgeglichener Test- und Trainingsdaten. Das Training und *Pruning* zugehöriger problemspezifischer Netztopologien ist in Abschnitt 6.3 und 6.5 beschrieben und ermöglicht die gleichzeitige, analytisch exakte Erfassung der Normalspannungen aller Stäbe des untersuchten Fachwerks in einem gemeinsamen Neuronalen Netz. Die zugehörige Vorverarbeitung der Test- und Trainingsdaten wird im folgenden Abschnitt vorgestellt.

6.1.4 Test- und Trainingsdaten

Die über den Permutationsalgorithmus des Programms FEA (siehe Abschnitt 5.3.2) erzeugten Trainingsdaten haben durch ihre Anordnung bzw. Reihenfolge der Präsentation gegenüber dem Netzwerk direkten Einfluß auf das Konvergenzverhalten der zu Training und *Pruning* eingesetzten Algorithmen. Die Abfolge der präsentierten Lernvektoren resultiert in einer bestimmten Spur des aktuellen Zustandsvektors der Netzübertragungsfunktion auf der Hyperfläche der Gesamtfehlerfunktion. Um negative Einfüsse aus einer monoton gleichen Abfolge der Lernvektoren auf Lernerfolg und Generalisierungsfähigkeit zu vermeiden, werden die einzelnen Muster der Trainingsdaten dem Netz zu jeder Epoche in jeweils zufälliger Reihenfolge präsentiert. Untersuchungen zu verschiedenen Verteilungscharakteristiken der Test- und Trainingsdaten im dimensionsbehafteten Ortsraum bzw. im dimensionslosen Bildraum bzgl. Lernerfolg und Generalisierungsfähigkeit des damit trainierten Netzwerks finden sich bei EMRICH [127]. Das beste Verhalten zeigt dort die äquidistante Verteilung der Muster im Ortsraum mit der zugehörigen selbstähnlichen, sich aus der Transformationsvorschrift der Dimensionsanalyse ergebenden Musterverteilung im Bildraum. Die in der vorliegenden Arbeit über den Permutationsalgorithmus erzeugten Test- und Trainingsdaten gehorchen dieser Musterverteilung.

Ein Vorteil aus der Datenvorverarbeitung durch Dimensionsanalyse ist die automatische Skalierung der Test- und Trainingsdaten auf numerisch ausgeglichene Werte. Hinsichtlich Konvergenzverhalten der verwendeten Lernalgorithmen Rprop und BackpropMomentum bzw.

Pruning-Algorithmus OBS haben sich drei bis vier Größenordnungen als maximaler Unterschied zwischen dem betragsmäßig größten und kleinsten Wert innerhalb der Test- und Trainingsdaten (z.B. das Intervall $]1 \cdot 10^{-2}, 1 \cdot 10^2]$) als sinnvoll erwiesen. Sollten die Werte als Ergebnis der Dimensionsanalyse überschritten werden, so können unter bestimmten Voraussetzungen Teile der Ein- und Ausgabevektoren durch einen gemeinsamen Skalierungsfaktor angepaßt werden (siehe Abschnitt 6.2). Die Beschränkung auf positive Werte ist aufgrund der Funktionsbelegung der Eingabeneuronen $\ln(in_i)$ für alle $i \in \{1, \dots, n\}$ der eingesetzten problemspezifischen Neuronalen Netze notwendig (siehe Abbildung 6).

Die Datenvorverarbeitung durch Dimensionsanalyse reduziert die Anzahl der zu einer vollständigen Beschreibung des untersuchten Zusammenhangs notwendigen Zahl der Kenngrößen der Problemstellung. Dies resultiert in einer Verringerung der Netzwerkgröße und des damit verbundenen Speicheraufwands für Test- und Trainingsdaten. Als angestrebte Folge sinkt der Rechenaufwand für Training und *Pruning* der Netzwerke.

Die folgenden Abschnitte beschreiben Untersuchungen zur Erfassung von Optimierungsrestriktionen (Knotenverschiebungen, Stabspannungen) durch problemspezifische Neuronale Netze. Hierbei werden normale Netztrainings- als auch *Pruning*-Verfahren eingesetzt.

6.2 Training problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Knotenverschiebung

In den folgenden Abschnitten sind Untersuchungen zur Beschreibung von Systemantworten ebener, statisch bestimmter Fachwerke durch Training und *Pruning* Neuronaler Netze beschrieben. Als Untersuchungsbeispiel dient das in Abbildung 11 dargestellte statisch bestimmte, zweidimensionale Fachwerk FW2.

Durch Training verschiedener, identisch in Neuronale Netze abgebildeter Ansatzfunktionen werden in diesem Abschnitt Möglichkeiten untersucht, die unter Einwirkung der äußeren Last P entstehenden Knotenverschiebungen des Fachwerks FW2 in Abbildung 11 analytisch exakt zu erfassen. Verschiedene Ansatzfunktionen repräsentieren dabei unterschiedliche Stufen von Vorabwissen zur untersuchten Problemstellung. Es folgen Abschnitte zur Datenvorverarbeitung mittels Dimensionsanalyse, zu den eingesetzten Ansatzfunktionen sowie zu Training und Diskussion der Ergebnisse.

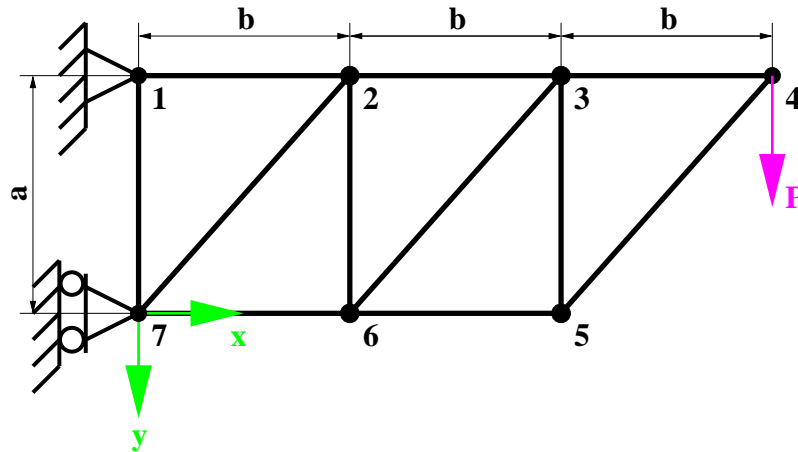


Abbildung 11: FW2: Statisch bestimmtes Fachwerk mit elf Stäben

6.2.1 Datenvorverarbeitung

Die Gesamttrainingsdaten sind in verschiedenen Kombinationen aus Einzeltrainingsdaten zu Fachwerken mit den Längenverhältnissen $\Psi = 0.8, 1, 2, 2.5$ für zwei verschiedene statische Lasten zusammengesetzt. Die Konstellationen verschiedener Ψ sollen hinsichtlich ihres Einflusses auf das Konvergenzverhalten und die Generalisierungsfähigkeit der mit diesen Daten trainierten Netzwerke untersucht werden. Ziel ist die analytisch exakte Erfassung aller im Gültigkeitsbereich des zugrundeliegenden physikalischen Modells möglichen Knotenverschiebungen beliebiger, zulässiger Längenverhältnisse Ψ . Gleichzeitig soll der notwendige Wertebereich der Trainingsdaten hinsichtlich Systemparametern und notwendiger Anzahl unterschiedlicher Ψ minimiert werden. Die dargestellte Beschreibung der Auslenkung $\delta_{\text{Kn } 6, y}$ des Knotens 6 des Fachwerks FW2 in y-Richtung ist repräsentativ und gilt in analoger Form für alle im folgenden untersuchten Knotenverschiebungen.

Die Dimensionsanalyse liefert über Relevanzliste und Dimensionstabelle folgende, den untersuchten Zusammenhang vollständig beschreibende dimensionslose Größen:

$$\begin{aligned} \Pi_1 = \frac{P}{S_{12}} \quad \Pi_2 = \frac{P}{S_{23}} \quad \Pi_3 = \frac{P}{S_{34}} \quad \Pi_4 = \frac{P}{S_{45}} \quad \Pi_5 = \frac{P}{S_{56}} \quad \Pi_6 = \frac{P}{S_{67}} \quad \Pi_7 = \frac{P}{S_{17}} \\ \Pi_8 = \frac{P}{S_{27}} \quad \Pi_9 = \frac{P}{S_{26}} \quad \Pi_{10} = \frac{P}{S_{36}} \quad \Pi_{11} = \frac{P}{S_{35}} \quad \Psi = \frac{a}{b} \quad \Delta_{i,y} = \frac{\delta_{\text{Kn } i,y}}{a} . \quad (109) \end{aligned}$$

Die Transformationsvorschriften zu Fachwerk FW2 sind für alle Knotenverschiebungen in y-Richtung mit $i = 1, \dots, 7$ gültig. Die Trainingsdaten ergeben sich über den Permutationsalgorithmus und die Transformationsvorschriften aus der Dimensionsanalyse in Gleichung (109) nach einer zusätzlichen Skalierung der Eingabe- und Sollausgabevektoren ohne die Geometriegrößen Ψ mittels eines konstanten Faktors. Der Eingabevektor \mathbf{in} sowie der Sollausgabe-

Trainingsdaten T_1, T_2, T_3		
Eingabevektor	in	$\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6, \Pi_7, \Pi_8, \Pi_9, \Pi_{10}, \Pi_{11}, \frac{1}{\Psi^3}, \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}}$
Sollausgabevektor	o	$\Delta_{6,y}$
Trainingsdaten	T_1	$\Psi = 1, 2, 2.5$ mit 12288 Mustern
	T_2	$\Psi = 0.8, 1, 2$ mit 12288 Mustern
	T_3	$\Psi = 1, 2$ mit 8192 Mustern

Tabelle 2: Trainingsdaten T_1, T_2, T_3 zu Fachwerk FW2 und ANT A, ANT B, ANT C

vektor **o** zur Beschreibung der Auslenkung des Knotens 6 in y-Richtung formen die Lernvektoren der Trainingsdaten. Durch Variation der Sollausgabegröße sind die Trainingsdaten zur Beschreibung aller Knotenverschiebungen des Fachwerks FW2 verwendbar. Die Anzahl der Trainingsmuster p folgt aus den Festlegungen der Fachwerksdefinitionsdatei über den Permutationsalgorithmus des Programms FEA. Beides ist in Tabelle 2 dargestellt.

6.2.2 Ansatzfunktionen

Anwendungsbeispiel des im folgenden beschriebenen Grundkonzepts sei die Verschiebung von Knoten 6 des Fachwerks FW2 in y-Richtung. Die beiden untersuchten Ansatzfunktionen beruhen auf dem Superpositionsprinzip des Einheitslastgesetzes und sind aus ihrer Konzeption heraus in der Lage, jeweils eine beliebige Knotenverschiebung des untersuchten Fachwerks FW2 in y-Richtung analytisch exakt zu erfassen. Die einzelnen Summanden der Ansatzfunktionen setzen sich aus dem Produkt eines Vorfaktors, einer dimensionslosen Π -Größe zur Berücksichtigung der Stabelastizitäten und der äußeren Last sowie aus dimensionslosen Geometriegrößen zusammen.

Die Geometriegröße $\left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}}$ für Diagonalstäbe und $\frac{1}{\Psi^3}$ für alle restlichen Stäbe der ersten Ansatzfunktion leiten sich aus den Erkenntnissen des Beispielfachwerks FW1 in Abschnitt 6.1.2 ab und berücksichtigen die geometrischen Eigenschaften der untersuchten Fachwerkstopologien unter den in Abschnitt 6.1.2 getroffenen Limitierungen vollständig. Ihre Zuordnung zu den einzelnen Summanden (Ansatzterme) erfolgt nach dem Vorbild der analytischen Lösung des Beispielfachwerks FW1 aus Abschnitt 6.1.3. Dies bedeutet eine Verknüpfung der Ansatzterme der Diagonalstäbe Stab_2 , Stab_3 und Stab_4 mit der Geometriegröße $\left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}}$ und der Ansatzterme aller restlichen Stäbe mit der Geometriegröße $\frac{1}{\Psi^3}$.

Zur Bestimmung der Variablen der Ansatzfunktionen (Verbindungsgewichte) werden die An-

satzfunktionen identisch in eine Neuronale Netztopologie abgebildet und deren variable Parameter durch Trainingsalgorithmen (siehe Abschnitt 6.3) über Simulationsrechnungen mit SNNS ermittelt. Gleichzeitig sieht die Systematik vor, unnötige Variablen der Ansatzfunktion durch Trainieren des zugehörigen Verbindungsgewichts auf null zu eliminieren. Der Aufbau der Netztopologie und die Belegung der Transferfunktionen zur Abbildung der Ansatzfunktionen entsprechen der in Abschnitt 5.2.3 beschriebenen Systematik. Jede dimensionslose, in der Netzansatzfunktion benötigte Beschreibungsgröße Π_i sowie die Geometrieterme werden dazu jeweils einem Neuron der Eingabeschicht, die dimensionslosen Zielgrößen der gesuchten Systemantworten jeweils einem Neuron der Ausgabeschicht zugeordnet.

Mit den Substitutionen $\gamma = \frac{1}{\Psi^3}$ und $\kappa = \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}}$ folgt die erste Ansatzfunktion in der Form

$$\begin{aligned} \Delta_{6,y} = & w_{14\ 25} (\Pi_1)^{w_{1\ 14}} (\gamma)^{w_{12\ 14}} + w_{15\ 25} (\Pi_2)^{w_{2\ 15}} (\gamma)^{w_{12\ 15}} + w_{16\ 25} (\Pi_3)^{w_{3\ 16}} (\gamma)^{w_{12\ 16}} + \\ & w_{17\ 25} (\Pi_4)^{w_{4\ 17}} (\kappa)^{w_{13\ 17}} + w_{18\ 25} (\Pi_5)^{w_{5\ 18}} (\gamma)^{w_{12\ 18}} + w_{19\ 25} (\Pi_6)^{w_{6\ 19}} (\gamma)^{w_{12\ 19}} + \\ & w_{20\ 25} (\Pi_7)^{w_{7\ 20}} (\gamma)^{w_{12\ 20}} + w_{21\ 25} (\Pi_8)^{w_{8\ 21}} (\kappa)^{w_{13\ 21}} + w_{22\ 25} (\Pi_9)^{w_{9\ 22}} (\gamma)^{w_{12\ 22}} + \\ & w_{23\ 25} (\Pi_{10})^{w_{10\ 23}} (\kappa)^{w_{13\ 23}} + w_{24\ 25} (\Pi_{11})^{w_{11\ 24}} (\gamma)^{w_{12\ 24}} . \end{aligned} \quad (110)$$

Die erste Ansatzfunktion in Gleichung (110) ist identisch in der Ausgangsnetztopologie ANT A in Abbildung 12 repräsentiert. Die zweite Ansatzfunktion

$$\Delta_{6,y} = \sum_{k=14}^{24} \left[w_{k\ 25} \left\{ \Pi_i^{w_{i\ k}} \left(\frac{1}{\Psi^3} \right)^{w_{12\ k}} \left(\left(1 + \frac{1}{\Psi^2} \right)^{\frac{3}{2}} \right)^{w_{13\ k}} \right\} \right] \quad (111)$$

mit $i = k - 13$ unterscheidet nicht mehr zwischen den Stäben des Fachwerks. Die Ansatzfunktion ist in die Ausgangsnetztopologie ANT B in Abbildung 13 abgebildet. Für beide Ansätze ist $\Delta_{6,y}$ dem Ausgabeneuron 25 zugeordnet. Die Ausgangsnetztopologien ANT A und ANT B sind in der Lage, jeweils eine beliebige Knotenverschiebung des untersuchten Fachwerks FW2 in y-Richtung unter den in Abschnitt 6.1.2 getroffenen Limitierungen analytisch exakt zu erfassen.

Die für die analytisch exakte Darstellung der Verschiebung des Knotens 6 in y-Richtung minimal notwendige Netztopologie NT C ist in Abbildung 14 dargestellt und dient der Kontrolle der untersuchten Systematik. Die zugehörige Netzübertragungsfunktion ist

$$\begin{aligned} \text{NF}_{\text{NT C}} = & w_{14\ 25} (\Pi_1)^{w_{1\ 14}} (\gamma)^{w_{12\ 14}} + w_{20\ 25} (\Pi_7)^{w_{7\ 20}} + w_{21\ 25} (\Pi_8)^{w_{8\ 21}} (\kappa)^{w_{13\ 21}} + \\ & w_{22\ 25} (\Pi_9)^{w_{9\ 22}} . \end{aligned} \quad (112)$$

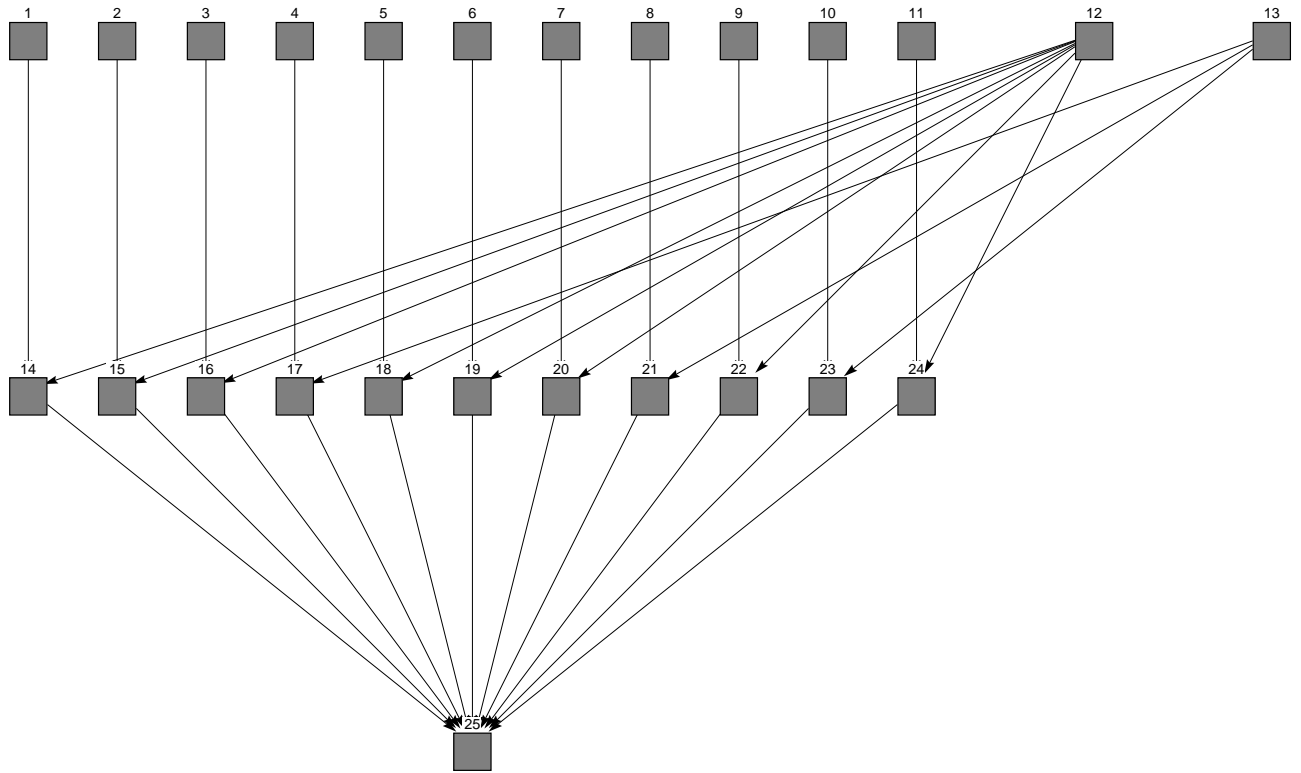


Abbildung 12: ANT A zur Ansatzfunktion in Gleichung (110)

6.2.3 Training und Ergebnisse

Die Grundeinstellungen der Trainingsrechnungen zu den Netzwerken ANT A, ANT B und NT C - jeweils mit den Trainingsdaten T_1, T_2, T_3 - sind in Tabelle 3 erfaßt. Die Simulationsrechnungen mit verschiedenen Variationen der Lernparameter dienen der Bestimmung einer ausreichend gleichmäßigen Verteilung der Trainingsdaten, die zur analytisch exakten Erfassung eines gesuchten, den Trainingsdaten inhärenten Zusammenhangs benötigt wird. Verteilung bezieht sich dabei auf die minimal notwendige Anzahl unterschiedlicher, charakteristischer Längenverhältnisse Ψ , die gemeinsam bei der Erzeugung der Trainingsdaten zu berücksichtigen sind. Die Elemente der Gewichtsmatrix einer problemspezifischen Netztopologie werden als Zielgewichte bezeichnet, wenn die Netzübertragungsfunktion exakt mit dem gesuchten Zusammenhang übereinstimmt. Die Ergebnisse der durchgeführten Simulationsrechnungen sind in Tabelle 4 zusammengefaßt. Die im folgenden verwendeten Tabellenwerte des Approximationsfehlers (AF) der Verbindungsgewichte stellen eine obere Schranke für den Approximationsfehler der im Training erzielten Annäherung der variablen Verbindungsgewichte w_{ij} an die Zielgewichte z_{ij} der analytisch exakten Lösung aus Gleichung (114) dar.

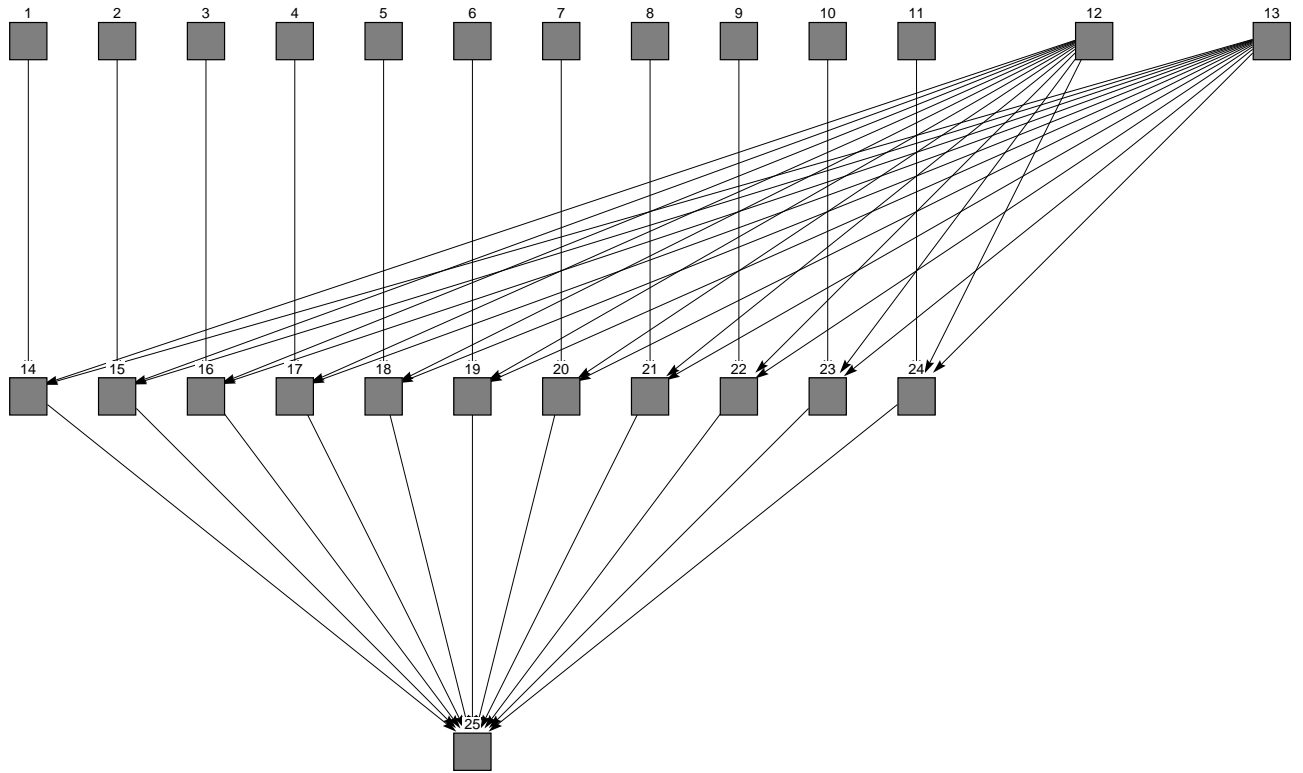


Abbildung 13: ANT B zur Ansatzfunktion in Gleichung (111)

Parameter Training ANT A, ANT B und NT C			
ANT/NT	A, B, C	Trainingsalgorithmus	BackpropMomentum
Trainingsdaten	T ₁ , T ₂ , T ₃	<i>Pruning</i> -Algorithmus	—
Testdaten	—	Initialisierungsintervall	[0.0, 1.0]

Tabelle 3: Parameter Training ANT A, ANT B und NT C

Der individuelle Approximationsfehler

$$AF = \left| \frac{z_{ij} - w_{ij}}{z_{ij}} \right| \quad (113)$$

ist für alle Gewichte des Netzwerks jeweils kleiner als der in Tabelle 4 gegebene Wert. Das Kürzel Ep steht für Epochen und beschreibt die Anzahl notwendiger Lernzyklen, bis SSE (siehe Gleichung (85)) den Wert 10^{-5} und damit die externe Genauigkeit des Simulators SNNS unterschreitet. Für die analytisch exakte Lösung der Verschiebung des Knotens 6 in y-Richtung in Fachwerk FW2 gilt in dimensionsloser Form

$$\Delta_{6,y} = \Pi_7 + \Pi_9 + \frac{3}{\Psi^3} \Pi_1 + \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}} \Pi_8. \quad (114)$$

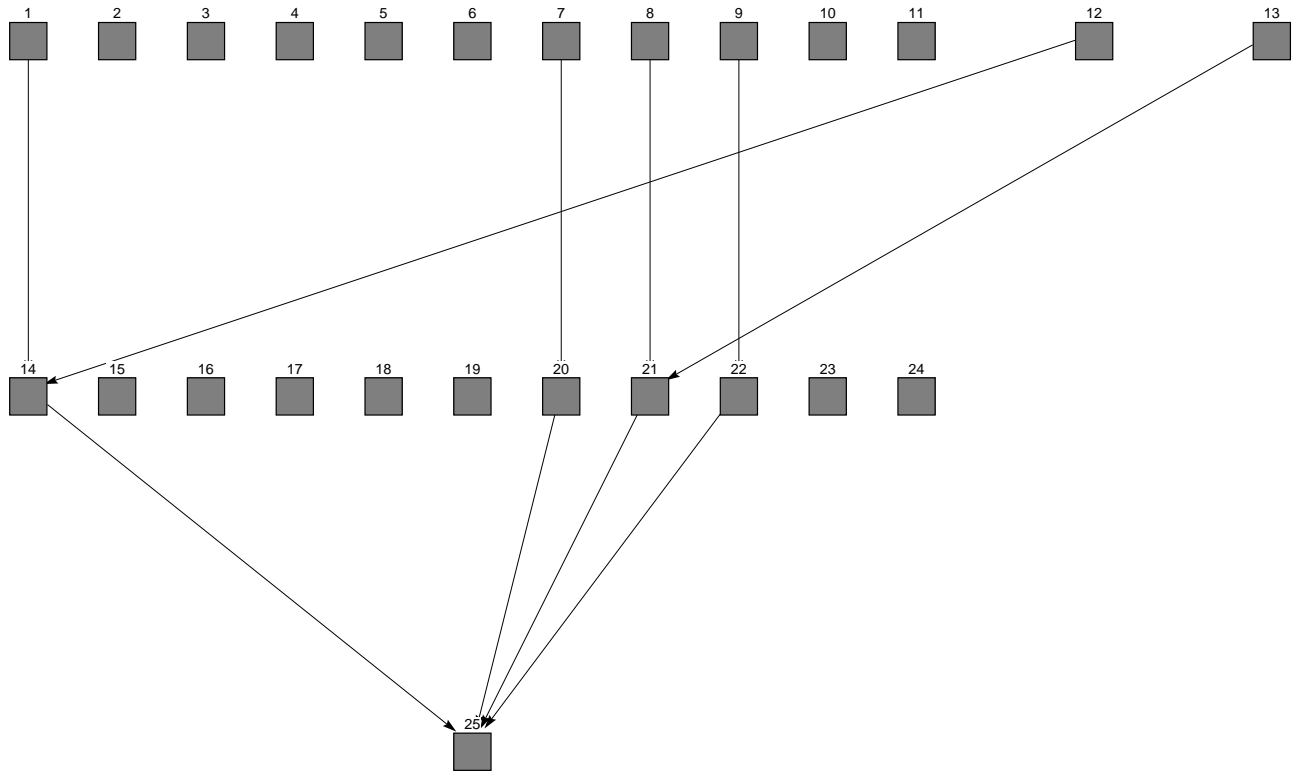


Abbildung 14: Minimale Netztopologie NT C zur Beschreibung von $\Delta_{6,y}$ in FW2

Ergebnisse Training ANT A, ANT B und NT C			
Trainingsdaten	ANT A	NT C	ANT B
T ₁ SSE < 10 ⁻⁵ nach AF <	14 000 Ep 10 ⁻¹	4000 Ep 10 ⁻²	— —
T ₂ SSE < 10 ⁻⁵ nach AF <	2000 Ep 10 ⁻²	4000 Ep 10 ⁻²	> 200 000 10 ⁻¹ aber 1 U
T ₃ SSE < 10 ⁻⁵ nach AF <	6000 Ep 10 ⁻²	6000 Ep 10 ⁻²	— —

Tabelle 4: Ergebnisse Training ANT A, ANT B und NT C

Die variablen Parameter des hierauf basierenden Ansatzes aus Gleichung (112) zeigen beim Training der zugehörigen Netztopologie NT C mit allen untersuchten Trainingsdaten einen maximalen Approximationsfehler von jeweils 10⁻² für alle Gewichte. Damit ist die grundsätzliche Funktionsfähigkeit dieser Vorgehensweise bestätigt.

Zur Ausgangsnetztopologie ANT A konnten die Zielgewichte für alle untersuchten Trainings-

daten mit einem maximalen Approximationsfehler von 10^{-1} bzw. 10^{-2} ermittelt werden. Aus dem Vergleich der gesuchten Lösung und der durch Training der Ausgangsnetztopologien realisierten Netzübertragungsfunktionen folgt, daß einige der Zielgewichte beliebige Werte $z_{ij} = \text{bel}$ annehmen können, da sie z.B. durch einen Vorfaktor gleich null keinen Einfluß auf den Funktionswert mehr haben. Ein Gewicht der Gewichtsmatrix der trainierten Ausgangsnetztopologie ANT B zeigt nach Training mit den Trainingsdaten T_2 keine Annäherung an den Wert des zugehörigen Zielgewichts (Kürzel 1 U).

Simulationsrechnungen zur Approximation der Verschiebung von Knoten 3 und 4 aus Fachwerk FW2 in y-Richtung zeigen analoges Verhalten. Trainingsdaten aus einem beschränkten Wertebereich des zu beschreibenden funktionalen Zusammenhangs (z.B. T_3 mit $\Psi = 1,2$) ermöglichen die Bestimmung eines trainierten Neuronalen Netzes mit maximaler Generalisierungsfähigkeit (Beispiel: Trainingsdaten T_3 mit ANT A). Gleichzeitig ergibt sich auch nach erheblicher Variation der Lernparameter eine deutliche Abhängigkeit der durch reines Training erzielbaren Approximationsergebnisse von der Komplexität der Ausgangsnetztopologie.

Die Ergebnisse des Abschnitts bauen auf den Untersuchungen der zu Beginn dieses Kapitels genannten Literaturstellen hinsichtlich der eingesetzten Datenvorverarbeitung durch Dimensionsanalyse sowie der Systematik zur Abbildung eines gegebenen Zusammenhangs in ein Neuronales Netz auf. Neu ist die Verwendung charakteristischer Längenverhältnisse zur Berücksichtigung unterschiedlicher geometrischer Verhältnisse der zu erfassenden Fachwerkstruktur. In der vorliegenden Arbeit ist durch eine Kombination verschiedener Längenverhältnisse die notwendige gleichmäßige Verteilung der Trainingsdaten zur gleichzeitigen Erfassung der Systemreaktion Knotenverschiebung mehrerer Fachwerksgeometrien in einem gemeinsamen Neuronalen Netz realisiert.

Die Möglichkeiten, die exakte Repräsentation eines Zusammenhangs durch reines Training zu ermitteln, dabei überflüssige Verbindungsgewichte auf null zu trainieren und diese damit aus der Ansatzfunktion zu eliminieren, sind beschränkt. Die Trainingsergebnisse zur Ausgangsnetztopologie ANT B, zugehörig zur Ansatzfunktion in Gleichung (111), zeigen die Grenzen der mit vertretbarem Aufwand möglichen Ermittlung exakter Zusammenhänge durch reines Training auf. Abhilfe bietet die Vereinfachung allgemeiner Ansatzfunktionen durch *Pruning* mit anschließendem Nachtraining der ausgedünnten Netztopologien. Untersuchungen zu diesem neuen Ansatz finden sich in den Abschnitten 6.4 und 6.5. Die in diesem Abschnitt gewonnenen Erkenntnisse erlauben die Erfassung jeweils einer Knotenverschiebung durch ein geeignetes, problemspezifisches Neuronales Netz. Die Erfassung aller Knotenverschiebungen in einem gemeinsamen Neuronalen Netz ist Ziel der Untersuchungen in Abschnitt 6.4.

6.3 Training problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Stabspannung

Im folgenden Abschnitt werden Möglichkeiten untersucht, die Einhaltung von Restriktionen in Optimierungsalgorithmen durch trainierte Neuronale Netze zu kontrollieren. Beispielrestriktion ist die Einhaltung maximal zulässiger Stabspannungen. Dies geschieht durch trainierte Neuronale Netze, die in der Lage sind, die aktuellen Spannungen aller Stäbe der untersuchten Struktur zu jedem Optimierungsschritt und über den gesamten Gültigkeitsbereich des zugrundeliegenden physikalischen Modells zu erfassen und exakt zu reproduzieren. Für die ständig notwendige Kontrolle der Restriktionen in jedem Iterationsschritt des Optimierungsverfahrens bietet die Systematik Vorteile aufgrund der kurzen Antwortzeiten trainierter Neuronaler Netze.

Die in diesem und Abschnitt 6.5 besprochenen Ansätze untersuchen verschiedene Ausgangsnetztopologien (Ansatzfunktionen), unterschiedliche Trainingsdaten sowie Training und *Pruning* problemspezifischer Netztopologien zur vollständigen Erfassung der den Trainingsdaten inhärenten Zusammenhänge. Anwendungsbeispiel ist das statisch bestimmte Fachwerk FW2 aus Abbildung 11 im Abschnitt 6.2. Es gelten die in Abschnitt 6.1.2 getroffenen Einschränkungen.

6.3.1 Datenvorverarbeitung

Die in diesem und in Abschnitt 6.5 eingesetzte Form der Datenvorverarbeitung basiert nicht auf Dimensionsanalyse. Die in Gleichung (115) definierte Datenvorverarbeitung ermöglicht aber gleichfalls eine für Training und *Pruning* Neuronaler Netze geeignete, numerisch ausgeglichene Darstellung der signifikanten Größen des untersuchten Zusammenhangs. Die gegebenen Transformationsvorschriften und die zugehörigen geometrischen Kenngrößen sind aus den Erkenntnissen zu Fachwerk FW1 in Abschnitt 6.1.3 abgeleitet.

$$\begin{aligned}
 \Theta_1 &= \frac{P}{A_n \sigma_n} & \Theta_2 &= \frac{A_{1\ 2}}{A_n} & \Theta_3 &= \frac{A_{2\ 3}}{A_n} & \Theta_4 &= \frac{A_{3\ 4}}{A_n} & \Theta_5 &= \frac{A_{4\ 5}}{A_n} & \Theta_6 &= \frac{A_{5\ 6}}{A_n} \\
 \Theta_7 &= \frac{A_{6\ 7}}{A_n} & \Theta_8 &= \frac{A_{1\ 7}}{A_n} & \Theta_9 &= \frac{A_{2\ 7}}{A_n} & \Theta_{10} &= \frac{A_{3\ 6}}{A_n} & \Theta_{11} &= \frac{A_{2\ 6}}{A_n} & \Theta_{12} &= \frac{A_{3\ 5}}{A_n} \\
 \Gamma_1 &= \frac{\sigma_{1\ 2}}{\sigma_n} & \Gamma_2 &= \frac{\sigma_{2\ 3}}{\sigma_n} & \Gamma_3 &= \frac{\sigma_{3\ 4}}{\sigma_n} & \Gamma_4 &= \frac{\sigma_{4\ 5}}{\sigma_n} & \Gamma_5 &= \frac{\sigma_{5\ 6}}{\sigma_n} & \Gamma_6 &= \frac{\sigma_{6\ 7}}{\sigma_n} \\
 \Gamma_7 &= \frac{\sigma_{1\ 7}}{\sigma_n} & \Gamma_8 &= \frac{\sigma_{2\ 7}}{\sigma_n} & \Gamma_9 &= \frac{\sigma_{3\ 6}}{\sigma_n} & \Gamma_{10} &= \frac{\sigma_{2\ 6}}{\sigma_n} & \Gamma_{11} &= \frac{\sigma_{3\ 5}}{\sigma_n} & \Psi &= \frac{a}{b}
 \end{aligned} \tag{115}$$

Die Rückrechnung der dimensionslosen Kenngrößen aus Gleichung (115) ist am Beispiel der Normalspannung im Stab_{4 5} in Fachwerk FW2 durch

$$\sigma_{4 5} = -\frac{P}{A_{4 5}} \frac{\sqrt{a^2 + b^2}}{a} \Leftrightarrow \Gamma_4 = -\frac{\Theta_1}{\Theta_5} \left(1 + \frac{1}{\Psi^2}\right)^{\frac{1}{2}} \quad (116)$$

demonstriert. Die sich aus der Kombination verschiedener Längenverhältnisse Ψ ergebenden Trainingsdaten T_4, T_5, T_6 sind in Tabelle 5 dargestellt.

Trainingsdaten T_4, T_5, T_6		
Eingabevektor	in	$\frac{\Theta_1}{\Theta_2}, \frac{\Theta_1}{\Theta_3}, \frac{\Theta_1}{\Theta_4}, \frac{\Theta_1}{\Theta_5}, \frac{\Theta_1}{\Theta_6}, \frac{\Theta_1}{\Theta_7}, \frac{\Theta_1}{\Theta_8}, \frac{\Theta_1}{\Theta_9}, \frac{\Theta_1}{\Theta_{10}}, \frac{\Theta_1}{\Theta_{11}}, \frac{\Theta_1}{\Theta_{12}}, \Psi^{-1}, \left(1 + \frac{1}{\Psi^2}\right)^{\frac{1}{2}}$
Sollausgabevektor	o	$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6, \Gamma_7, \Gamma_8, \Gamma_9, \Gamma_{10}, \Gamma_{11}$
Trainingsdaten	T_4	$\Psi = 1$ mit 4096 Mustern
	T_5	$\Psi = 1, 2$ mit 8192 Mustern
	T_6	$\Psi = 0.8, 1, 2$ mit 12288 Mustern

Tabelle 5: Trainingsdaten T_4, T_5, T_6 zu Fachwerk FW2 und NT D

6.3.2 Ansatzfunktionen

Zu jedem Stab des untersuchten Fachwerks FW2 ist jeweils eine der Einzelansatzfunktionen

$$\begin{aligned}
 \Gamma_1 &= w_{14 25} \left(\frac{\Theta_1}{\Theta_2}\right)^{w_{1 14}} (\Psi^{-1})^{w_{12 14}} & \Gamma_2 &= w_{15 26} \left(\frac{\Theta_1}{\Theta_3}\right)^{w_{2 15}} (\Psi^{-1})^{w_{12 15}} \\
 \Gamma_3 &= w_{16 27} \left(\frac{\Theta_1}{\Theta_4}\right)^{w_{3 16}} (\Psi^{-1})^{w_{12 16}} & \Gamma_4 &= w_{17 28} \left(\frac{\Theta_1}{\Theta_5}\right)^{w_{4 17}} (\xi)^{w_{13 17}} \\
 \Gamma_5 &= w_{18 29} \left(\frac{\Theta_1}{\Theta_6}\right)^{w_{5 18}} (\Psi^{-1})^{w_{12 18}} & \Gamma_6 &= w_{19 30} \left(\frac{\Theta_1}{\Theta_7}\right)^{w_{6 19}} (\Psi^{-1})^{w_{12 19}} \\
 \Gamma_7 &= w_{20 31} \left(\frac{\Theta_1}{\Theta_8}\right)^{w_{7 20}} & \Gamma_8 &= w_{21 32} \left(\frac{\Theta_1}{\Theta_9}\right)^{w_{8 21}} (\xi)^{w_{13 21}} \\
 \Gamma_9 &= w_{22 33} \left(\frac{\Theta_1}{\Theta_{10}}\right)^{w_{9 22}} (\xi)^{w_{13 22}} & \Gamma_{10} &= w_{23 34} \left(\frac{\Theta_1}{\Theta_{11}}\right)^{w_{10 23}} \\
 \Gamma_{11} &= w_{24 35} \left(\frac{\Theta_1}{\Theta_{12}}\right)^{w_{11 24}} & &
 \end{aligned} \quad (117)$$

mit $\xi = \left(1 + \frac{1}{\Psi^2}\right)^{\frac{1}{2}}$ identisch durch die Methodik problemspezifischer Neuronaler Netze in die Netztopologie NT D (Abbildung 15) abgebildet. Die Zuordnung der Eingabe- und Sollausgabevektoren der Trainingsdaten zu den Neuronen der Ein- und Ausgabeschicht der Netztopologie NT D entspricht jeweils der Anordnung in Tabelle 5. Die Γ_r zu Stab r sind jeweils

dem Ausgabeneuron j mit $j = r + 24$ zugeordnet. Die Bestimmung der variablen Parameter w_{ij} der Ansätze erfolgt über Trainingsverfahren. Ein Vergleich mit der analytisch exakten Lösung ermöglicht die Validierung der Vorgehensweise und der damit erzielten Ergebnisse.

Die Netztopologie NT D stellt die minimal notwendige Netztopologie zur analytisch exakten Beschreibung der Normalspannungen aller Stäbe des Fachwerks FW2 in einem gemeinsamen Neuronalen Netz zugehörig zu den Ansatzfunktionen aus Gleichung (117) dar. Ihr Aufbau ist aus der Lösungssystematik der Ergebnisse von Fachwerk FW1 (siehe Abschnitt 6.1.2) abgeleitet. Allgemeinere Ansatzfunktionen zu der Problemstellung werden in Abschnitt 6.5 untersucht.

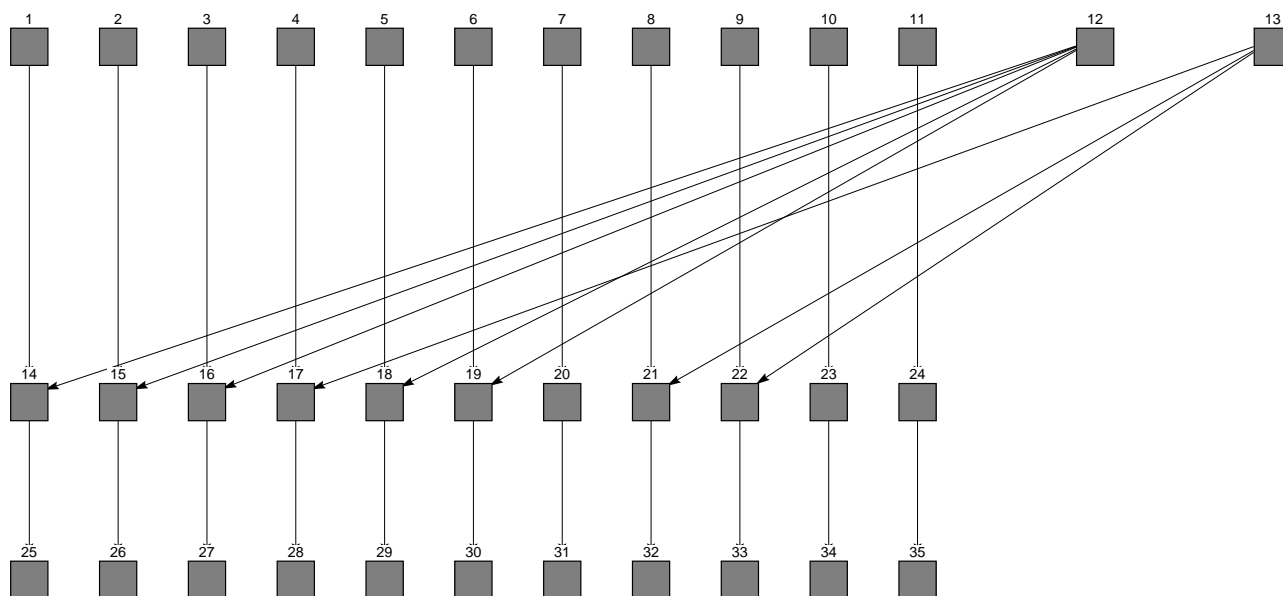


Abbildung 15: NT D zu den Ansatzfunktionen in Gleichung (117)

6.3.3 Training und Ergebnisse

Die beschriebenen Untersuchungen und Ergebnisse basieren auf reinem Training der Netztopologie NT D mit den Trainingsdaten T_4, T_5, T_6 . Die verwendeten Einstellungen zum Training sind in Tabelle 6 und die daraus resultierenden Ergebnisse in Tabelle 7 zusammengefaßt. Das Training der Netztopologie NT D mit den Trainingsdaten T_5 oder T_6 bestätigt die getroffenen Annahmen. Der erreichte individuelle Approximationsfehler AF nach Gleichung (113) ist für alle Gewichte der trainierten Netztopologie NT D jeweils kleiner als 10^{-4} . Die mit nur einem Längenverhältnis $\Psi = 1$ erzeugten Trainingsdaten T_4 sind nicht in der Lage, dem Lern-

Parameter Training NT D			
NT	D	Trainingsalgorithmus	BackpropMomentum
Trainingsdaten	T ₄ , T ₅ , T ₆	<i>Pruning</i> -Algorithmus	—
Testdaten	—	Initialisierungsintervall	[-1.0, 1.0]

Tabelle 6: Parameter Training NT D

Ergebnisse Training NT D	
Trainingsdaten	NT D
T ₄ SSE < 10 ⁻⁵ nach AF <	< 1000 Ep 10 ⁻² aber 7 U
T ₅ SSE < 10 ⁻⁵ nach AF <	< 1000 Ep 10 ⁻⁴
T ₆ SSE < 10 ⁻⁵ nach AF <	< 1000 Ep 10 ⁻⁴

Tabelle 7: Ergebnisse Training NT D

algorithmus ausreichende Informationen zur Bestimmung aller Zielgewichte zur Verfügung zu stellen. Grund ist die in den Trainingsdaten T₄ fehlende Variation der charakteristischen Längenabmessungen der Stäbe. Der Netzfehler sinkt bei allen untersuchten Trainingsdaten für die dargestellten Fälle nach maximal 1000 Epochen unter $SSE = 10^{-5}$.

Die entwickelte Methodik ist in der Lage, die Normalspannungen **aller** Stäbe eines Fachwerks in **einem gemeinsamen** Neuronalen Netz analytisch exakt zu erfassen. Dies wird durch eine geeignete Datenvorverarbeitung ohne Dimensionsanalyse sowie durch die Berücksichtigung von Einzelansatzfunktionen zu jedem Stab des zu erfassenden Fachwerks und deren Kombination in einem gemeinsamen Neuronalen Netz möglich. In den Test- und Trainingsdaten werden charakteristische Längenverhältnisse zur Berücksichtigung unterschiedlicher geometrischer Verhältnisse der zu erfassenden Fachwerkstruktur eingesetzt. Durch die Kombination verschiedener Längenverhältnisse wird die notwendige, gleichmäßige Verteilung der Trainingsdaten zur gleichzeitigen Erfassung der Normalspannungen aller Stäbe des untersuchten Fachwerks realisiert. Die erzielten Ergebnisse basieren auf Ansatzfunktionen, die der Funktionenklasse der untersuchten Problemstellung sehr nahe kommen. Untersuchungen zu allgemeineren Approximationsansätzen und damit geringerem, notwendigem Kenntnisstand zur Problemstellung folgen in Abschnitt 6.5.

6.4 Pruning problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Knotenverschiebung

In diesem Abschnitt wird das *Pruning* problemspezifischer Ausgangsnetztopologien mit anschließendem Nachtraining der ausgedünnten Netztopologie untersucht. Als Ansatzfunktion kommt der in der vorliegenden Arbeit entwickelte und in Abschnitt 5.2.4 vorgestellte, multivariable Summen-Produkt-Ansatz zur Anwendung. Eine erfolgreich und damit maximal reduzierte Netztopologie beschreibt den in den Trainingsdaten enthaltenen Zusammenhang innerhalb des Gültigkeitsbereichs des zugrundeliegenden physikalischen Modells vollständig. Das Neuronale Netz besitzt somit die maximal mögliche Generalisierungsfähigkeit. Dies ergibt sich zum einen aus der Existenz eines analytisch exakten Zusammenhangs zur untersuchten Problemstellung und zum anderen aus dem Generalisierungspotential der dimensionslosen Π -Größen als Resultat der verwendeten Dimensionsanalyse. Anwendungsbeispiel ist die Reduktion einer allgemeinen Ansatzfunktion (ANT) auf den zur analytisch exakten Beschreibung aller Knotenverschiebungen des Fachwerks FW2 in y -Richtung minimal notwendigen Zusammenhang (ausgedünnte Netztopologie).

Die verwendete Schar von Ansatzfunktionen basiert auf den Überlegungen des Einheitslastgesetzes. Aus den Erkenntnissen in Abschnitt 6.2 ist jede einzelne der Ansatzfunktionen in der Lage, jeweils eine beliebige, dimensionslose Knotenverschiebung des untersuchten Fachwerks FW2 analytisch exakt zu erfassen. Es gelten die in Abschnitt 6.1.2 getroffenen Limitierungen. Die Ansatzfunktionen

$$\Delta_{q,y} = \sum_{k=14}^{24} \left[w_{k,j} \left\{ \Pi_i^{w_{i,k}} \left(\frac{1}{\Psi^3} \right)^{w_{12,k}} \left(\left(1 + \frac{1}{\Psi^2} \right)^{\frac{3}{2}} \right)^{w_{13,k}} \right\} \right] \text{ für alle } q = 1, \dots, 7 \quad (118)$$

mit $i = k - 13$ und $j = q + 24$ ermöglichen zusammengenommen die Erfassung **aller** sieben Knotenverschiebungen des Fachwerks FW2 in y -Richtung in einem gemeinsamen Neuronalen Netz.

Die Ansatzfunktionen aus Gleichung (118) sind über die Zuordnung der dimensionslosen Knotenverschiebung $\Delta_{q,y}$ zum Ausgabeneuron $j = q + 24$ identisch in die Ausgangsnetztopologie ANTE in Abbildung 16 abgebildet.

Mit reinem *Pruning* kann für die untersuchte Problemstellung im Rahmen von Vorversuchen keine ausreichende Annäherung der Verbindungsgewichte an ihre jeweiligen Zielgewichte erzielt werden. Die Ermittlung der Variablen der identisch in einem Neuronalen Netz repräsentierten Ansatzfunktionen erfolgt deshalb in zwei Schritten:

- Der *Pruning*-Algorithmus OBS eliminiert unnötige Verbindungen und verdeckte Neu-

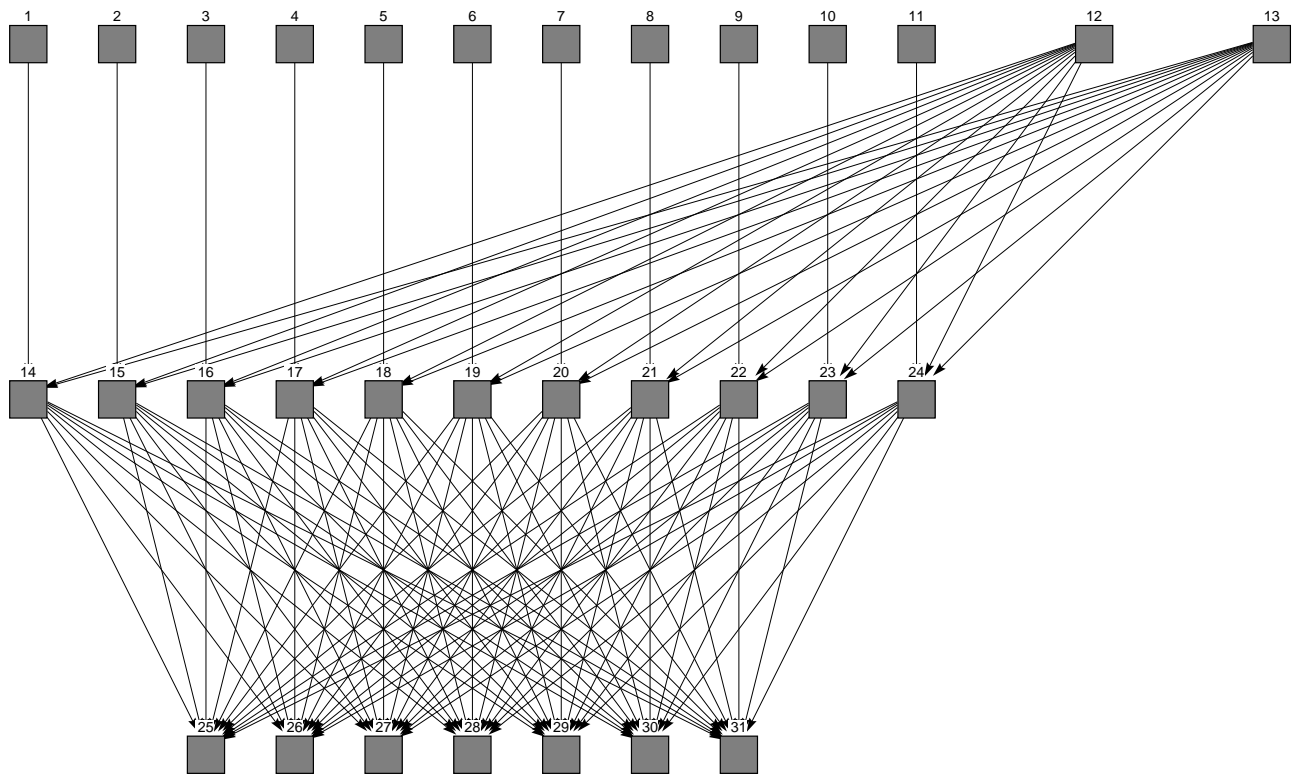


Abbildung 16: ANT E zu den Ansatzfunktionen in Gleichung (118)

ronen und damit unnötige Variablen bzw. Terme der Ansatzfunktion.

- Die restlichen Verbindungsgewichte (Restvariablen) der ausgedünnten Netztopologie werden nach erneuter zufälliger Initialisierung über reines Training (Nachtraining) an den inhärenten Zusammenhang der Trainingsdaten angepaßt.

Die Datenvorverarbeitung erfolgt durch Dimensionsanalyse gemäß den Transformationsvorschriften aus Gleichung (109). Der Aufbau der sich ergebenden Trainingsdaten ist in Tabelle 8,

Trainingsdaten T_7, T_8		
Eingabevektor	in	$\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6, \Pi_7, \Pi_8, \Pi_9, \Pi_{10}, \Pi_{11}, \frac{1}{\Psi^3}, \left(1 + \frac{1}{\Psi^2}\right)^{\frac{3}{2}}$
Sollausgabevektor	o	$\Delta_{1,y}, \Delta_{2,y}, \Delta_{3,y}, \Delta_{4,y}, \Delta_{5,y}, \Delta_{6,y}, \Delta_{7,y}$
Trainingsdaten	T_7	$\Psi = 1, 2$ mit 8192 Mustern
	T_8	$\Psi = 0.8, 1, 2$ mit 12288 Mustern

Tabelle 8: Trainingsdaten T_7, T_8 zu Fachwerk FW2 und ANT E

die Parametereinstellungen der durchgeführten Simulationsrechnungen sind in Tabelle 9 zusammengestellt. Hierbei bezeichnet z.B. die Abkürzung „OBS mit Rprop“ den Einsatz des *Pruning*-Verfahrens OBS mit dem untergeordneten Lernverfahren Rprop.

Parameter Pruning und Training ANT E			
ANT	E	Nachtrainingsalgorithmen	BackpropMomentum und Rprop
Trainingsdaten	T ₇ , T ₈	<i>Pruning</i> -Algorithmen	OBS mit BackpropMomentum und Rprop MBP mit BackpropMomentum und Rprop
Testdaten	—	Initialisierungsintervall	[0.0, 1.0]

Tabelle 9: Parameter *Pruning* und Training ANT E

Das *Pruning*-Verfahren OBS mit dem untergeordneten Lernverfahren BackpropMomentum zeigt das größte Reduktionspotential der untersuchten Verfahren. Die dabei erzielte Ausdünnung der Ausgangsnetztopologie ANT E beschränkt sich auf unnötige, hinsichtlich des zu beschreibenden Zusammenhangs überflüssige Variablen und Terme der Ansatzfunktionen. Ist dies erfolgreich durchgeführt, so besteht die Netzübertragungsfunktion der ausgedünnten Resttopologie nur aus Variablen und Termen, die zur analytisch exakten Erfassung des gesuchten Zusammenhangs notwendig sind.

Das Ergebnis der Simulationsrechnung okt53o (siehe Abbildung 17) stellt die minimal notwendige Netztopologie zur gleichzeitigen, analytisch exakten Beschreibung aller Knotenverschiebungen des Fachwerks FW2 in y-Richtung dar. Hierzu eingesetztes Verfahren ist OBS mit dem untergeordneten Lernverfahren BackpropMomentum. Knoten 1 in Fachwerk FW2 ist ein Festlager und damit unverschieblich. Dies erklärt den Umstand, daß die Ausgabeinheit 25 der reduzierten Netztopologie NT okt53o in Abbildung 17 gegenüber der restlichen Netztopologie isoliert ist. Mathematisch gesehen sind damit alle für eine analytisch exakte Beschreibung des zugrundeliegenden physikalischen Modells nicht notwendigen Variablen und Terme der Ansatzfunktionen aus Gleichung (118) gelöscht. Die dabei als direktes Resultat des *Pruning*-Verfahrens ermittelten Verbindungsgewichte sind nur unzureichend oder gar nicht an ihre jeweiligen Zielgewichte angenähert. Die exakte Ermittlung des gesuchten Zusammenhangs erfolgt deshalb durch Nachtraining der optimal ausgedünnten Netztopologie.

Das *Pruning*-Verfahren OBS mit Rprop als untergeordnetem Lernverfahren erzielt keine vergleichbaren Ergebnisse. Das *Pruning*-Verfahren MBP zeigt für die untersuchten Parametervariationen weder mit BackpropMomentum noch mit Rprop als untergeordnetem Lernverfahren verwertbare Resultate. OBS mit Rprop und MBP sind nicht in der Lage, die dem Training und

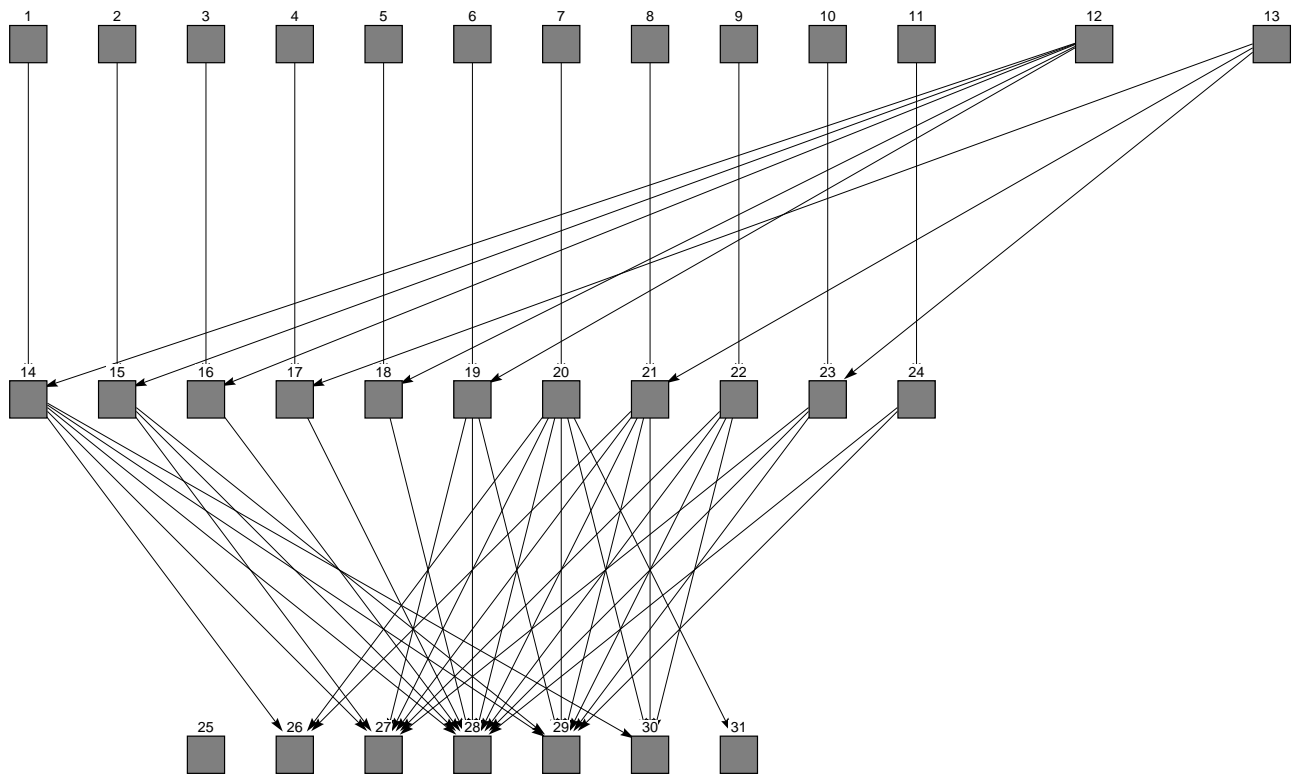


Abbildung 17: Reduzierte Netztopologie NT okt53o

Pruning Neuronaler Netze zugrunde liegende Minimierungsaufgabe zu lösen.

Die durch Nachtraining der Netztopologie NT okt53o erzielten Ergebnisse sind in Tabelle 10 erfaßt. Die Restgewichte der ausgedünnten Netztopologie, die durch das *Pruning*-Verfahren

Ergebnisse Nachtraining reduzierte Netztopologie NT okt53o		
Trainingsdaten	BackpropMomentum	Rprop
T ₇ SSE < 10 ⁻⁵ nach AF <	26 000 Ep 10 ⁻²	— —
T ₈ SSE < 10 ⁻⁵ nach AF <	41 000 Ep 10 ⁻³	— —

Tabelle 10: Ergebnisse Nachtraining reduzierte Netztopologie NT okt53o

nicht oder nur unzureichend an ihre jeweiligen Zielgewichte angenähert werden, sind durch Nachtraining nach gleichverteilter Initialisierung korrekt bestimmbar. Dies zeigen die Ergebnisse zum Nachtraining mit dem Lernverfahren BackpropMomentum. Für das Lernverfahren Rprop kann dagegen über das gesamte untersuchte Spektrum der Lernparameter keine signi-

fikante Annäherung der Rest- an die Zielgewichte festgestellt werden.

Der *Pruning*-Algorithmus OBS zeigt für die untersuchte Aufgabenstellung signifikant bessere Ergebnisse bzgl. Generalisierungsleistung und Reduktionspotential als das ebenfalls untersuchte MBP-Verfahren (siehe Abschnitt 5.5.3 und SCHREINER [116]). Für den Fall eines in die Ausgangsnetztopologie implementierten, geeigneten problemspezifischen Funktionsansatzes kann die optimal reduzierte (minimale) und vollständig nachtrainierte Netztopologie den in den Trainingsdaten implizit enthaltenen Zusammenhang exakt erfassen. Sie zeigt dann innerhalb des Definitionsbereichs des zugrundeliegenden physikalischen Modells die maximale Generalisierungsfähigkeit.

Zur ausreichenden Verteilung sind die Trainingsdaten wieder aus Daten zu Fachwerken mit verschiedenen charakteristischen Längenverhältnissen kombiniert. Aufbauend auf den Resultaten der vorangegangenen Abschnitte ist die im vorliegenden Abschnitt entwickelte Methode in der Lage, die minimal notwendige Netztopologie zur Erfassung aller Knotenverschiebungen des untersuchten Fachwerks in eine Koordinatenrichtung zu ermitteln.

Die erzielten Ergebnisse ermöglichen die exakte Beschreibung der Optimierungsrestriktion Knotenverschiebung durch geeignete Neuronale Netze. Im nächsten Abschnitt folgen analoge Untersuchungen zur Erfassung der Optimierungsrestriktion Stabspannung.

6.5 Pruning problemspezifischer Neuronaler Netze zur Erfassung der Systemreaktion Stabspannung

Der folgende Abschnitt untersucht Möglichkeiten, die Optimierungsrestriktion Stabspannung bzw. Materialfestigkeit durch *Pruning* sehr allgemeiner Ansatzfunktionen (Ausgangsnetztopologien) zu beschreiben. Ziel ist die analytisch exakte Erfassung der Normalspannungen aller Stäbe des Fachwerks FW2 in einem gemeinsamen, problemspezifischen Neuronalen Netz. Ein *Pruning*-Algorithmus reduziert die in problemspezifischen Neuronalen Netzen repräsentierten Ansatzfunktionen durch Löschen von unnötigen Variablen und Terme. Aus den Erfahrungen des vorhergehenden Abschnitts wird in diesem Abschnitt nur noch das *Pruning*-Verfahren OBS mit BackpropMomentum als untergeordnetem Lernverfahren eingesetzt.

Die untersuchten Ansatzfunktionen bestehen aus zwei Scharen von je 11 Teilansatzfunktionen. Der erste Ansatz kombiniert die in Abschnitt 6.3 verwendeten geometrischen Kenngrößen für alle Stäbe und verallgemeinert so den dortigen Funktionsansatz. Der zweite Ansatz gibt jeder der 11 Teilansatzfunktionen hinsichtlich mathematischem Aufbau und angesetzten Variablen das Potential, jeweils die Normalspannung eines beliebigen Stabs des untersuchten

Fachwerks FW2 analytisch exakt erfassen zu können.

Die erste Schar von Ansatzfunktionen für die dimensionslose Normalspannung des Stabs r im Fachwerk FW2 hat mit $i = r$, $k = r + 13$ und $j = r + 24$ die Form

$$\Gamma_r = w_{k j} \left[\left(\frac{\Theta_1}{\Theta_{r+1}} \right)^{w_{i k}} (\Psi^{-1})^{w_{12 k}} \left\{ \left(1 + \frac{1}{\Psi^2} \right)^{\frac{1}{2}} \right\}^{w_{13 k}} \right] \text{ für alle } r = 1, \dots, 11. \quad (119)$$

Die gewählte Schar von Ansatzfunktionen entspricht dem Lösungsmodell für die Bestimmung von Normalspannungen in Stabtragwerken. Die Ansatzfunktionen sind über die in Tabelle 5 gegebene Zuordnung gemeinsam in der Ausgangsnetztopologie ANT F in Abbildung 18 repräsentiert. Die verwendeten Variablen ergeben sich aus der Datenvorverarbeitung über die Transformationsvorschriften in Gleichung (115). Die zweite Schar von Ansatzfunktionen für

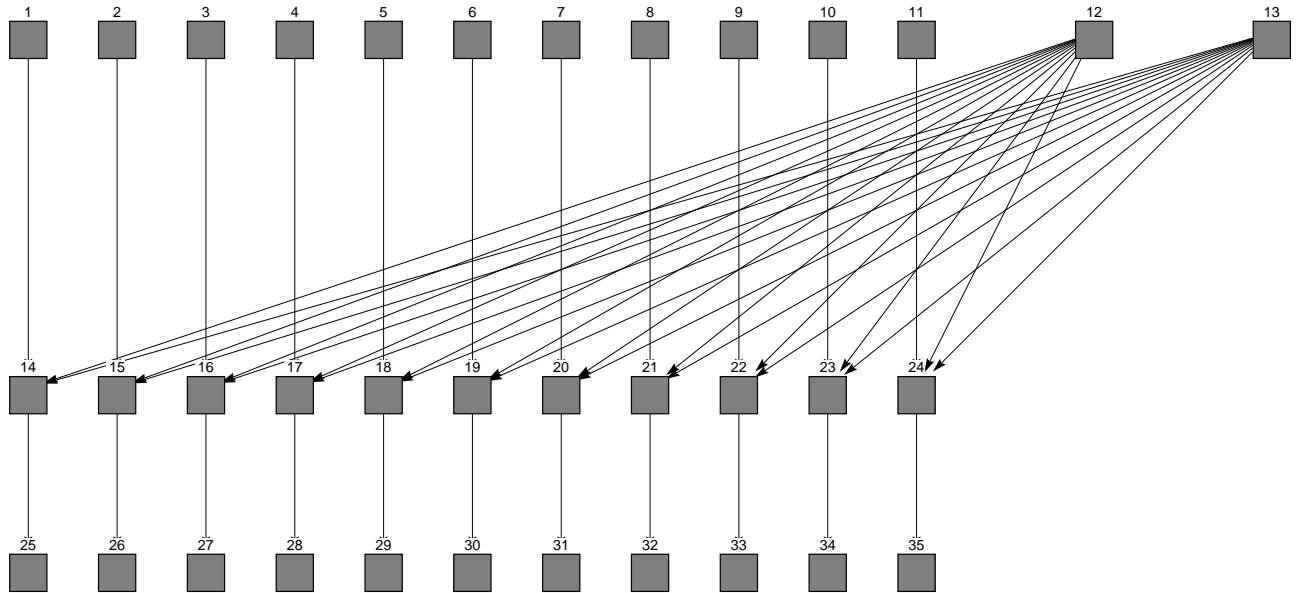


Abbildung 18: ANT F zu den Ansatzfunktionen in Gleichung (119)

die dimensionslose Normalspannung des Stabs r im Fachwerk FW2 hat mit $k = r + 13$ und $j = r + 24$ die Form

$$\Gamma_r = w_{k j} \left[\prod_{i=1}^{11} \left\{ \left(\frac{\Theta_1}{\Theta_{r+1}} \right)^{w_{i k}} \right\} (\Psi^{-1})^{w_{12 k}} \left\{ \left(1 + \frac{1}{\Psi^2} \right)^{\frac{1}{2}} \right\}^{w_{13 k}} \right] \text{ für alle } r = 1, \dots, 11. \quad (120)$$

Die Ansatzfunktionen sind über die in Tabelle 5 gegebene Zuordnung in der gemeinsamen Ausgangsnetztopologie ANT G in Abbildung 19 repräsentiert. Jede Teilansatzfunktion aus Gleichung (120) ist potentiell in der Lage, die Normalspannung eines beliebigen Stabs des untersuchten Fachwerks FW2 analytisch exakt zu erfassen.

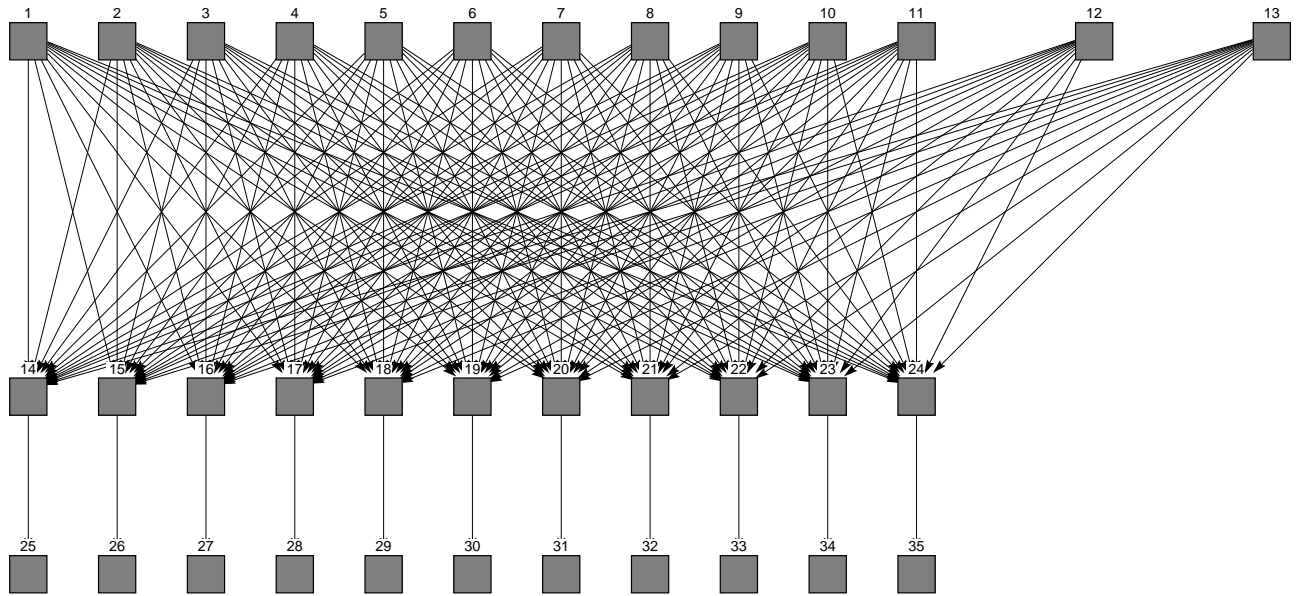


Abbildung 19: ANT G zu den Ansatzfunktionen in Gleichung (120)

Die beim *Pruning* der Ausgangsnetztopologien ANT F und ANT G verwendeten Trainingsdaten T_4, T_5, T_6 entsprechen den Angaben in Tabelle 5 aus Abschnitt 6.3.1. Die zur dimensionslosen, numerisch ausgeglichenen Erfassung des untersuchten physikalischen Vorgangs in den Trainingsdaten notwendige Datenvorverarbeitung erfolgt über die in Gleichung (115) gegebenen Transformationsvorschriften. Die Parametereinstellungen der Simulationsrechnungen sind in Tabelle 11 und zugehörige Ergebnisse in Tabelle 12 zusammengefaßt. Die Abkürzung

Parameter <i>Pruning</i> ANT F und ANT G			
ANT	F, G	Nachtrainingsalgorithmen	—
Trainingsdaten	T_4, T_5, T_6	<i>Pruning</i> -Algorithmen	OBS mit BackpropMomentum
Testdaten	—	Initialisierungsintervall	[0.0, 1.0]

Tabelle 11: Parameter *Pruning* ANT F und ANT G

„min. Netztop. erreicht“ kennzeichnet, ob die zur analytisch exakten Beschreibung der Normalspannungen aller Stäbe des Fachwerks FW2 in einem gemeinsamen Neuronalen Netz minimal notwendige Netztopologie erreicht wurde. Diese Netztopologie ist bereits als Netzwerk NT D in Abbildung 15 dargestellt.

Die Parametervariationen zum *Pruning* der Ausgangsnetztopologien ANT F und ANT G mit den Trainingsdaten T_4 und T_5 sind nicht in der Lage, die Netztopologie NT D zu ermitteln.

Ergebnisse Pruning ANT F und ANT G		
Trainingsdaten	ANT F	ANT G
T ₄ min. Netztop. erreicht	nein	nein
T ₅ min. Netztop. erreicht	nein	nein
T ₆ min. Netztop. erreicht AF <	ja 10 ⁻⁵	ja 10 ⁻⁴

Tabelle 12: Ergebnisse *Pruning* ANT F und ANT G

Analoge Parametervariationen mit den Trainingsdaten T₆ bestimmen dagegen die Netztopologie NT D exakt. Die für beide Ausgangsnetztopologien ANT F und ANT G als direktes Resultat des *Pruning*-Algorithmus' erzielten Approximationsfehler der Verbindungsgewichte sind jeweils so klein, daß ein Nachtraining der ausgedünnten Netztopologie überflüssig ist. Erst die Variation des Längenverhältnisses Ψ mit drei verschiedenen Werten in T₆ ermöglicht die optimale Reduktion der Ansatzfunktionen und gleichzeitige Bestimmung der variablen Parameter der Restterme. Dies ist durch die Beschränkungen der Rechengenauigkeit der Computer und Simulationssoftware begründet.

Um eine ausreichende Verteilung der Trainingsdaten sicherzustellen, sind diese aus einer Kombination von Daten zu Fachwerken mit unterschiedlichen geometrischen Kenngrößen zusammengesetzt. Die eingesetzte Datenvorverarbeitung basiert nicht auf der Dimensionsanalyse, sondern auf dem in Abschnitt 6.3 entwickelten Verfahren. Aufbauend auf den Resultaten der vorangegangenen Abschnitte ist die vorgestellte und untersuchte Systematik in der Lage, die minimal notwendige Netztopologie zur gleichzeitigen, analytisch exakten Erfassung aller Normalspannungen der Stäbe des untersuchten Fachwerks zu ermitteln.

6.6 Fazit

Die Untersuchungen und Ergebnisse des Kapitels 6 zeigen, daß die Beschreibung von Systemantworten bzw. Optimierungsrestriktionen statisch bestimmter Fachwerke durch trainierte, problemspezifische Neuronale Netze möglich ist.

Ausgehend von identisch in Neuronale Netze abgebildeten Ansatzfunktionen können damit sowohl alle Knotenverschiebungen als auch alle Normalspannungen des untersuchten Fachwerks analytisch exakt jeweils in einem gemeinsamen Neuronalen Netz erfaßt werden. Dies bedeutet eine maximale und damit eine auf dem gesamten Definitionsbereich des zugrundelie-

genden physikalischen Modells gültige Generalisierungsfähigkeit der jeweiligen Neuronalen Netze.

Die Erfassung von Systemantworten ebener Fachwerke (Knotenverschiebungen, Stabspannungen) durch problemspezifische Neuronale Netze ist als Resultat der vorliegenden Arbeit unter den in Abschnitt 6.1.2 getroffenen Annahmen möglich. Allgemeine Ansatzfunktionen, die identisch in Neuronale Netze abgebildet werden, können durch *Pruning* der Ausgangsnetztopologien auf die minimal notwendigen Netztopologien und damit die Ansatzfunktionen auf die minimal notwendigen Zusammenhänge reduziert werden.

Das folgende Kapitel untersucht eine Optimierungsaufgabe mit mehreren Optimierungskriterien und Restriktionen. Das Optimierungskriterium Versagenswahrscheinlichkeit und die Restriktionen maximal zulässige Knotenverschiebungen sowie Stabspannungen werden aus den Erkenntnissen dieses und der vorangegangenen Kapitel durch geeignete neuronale Netze beschrieben und in den Optimierungsalgorithmus integriert.

7 Neuronale Netze und Evolutionsstrategien zur Untersuchung von Mehrzieloptimierungsaufgaben

Grundidee der in diesem Kapitel vorgeschlagenen und untersuchten Methode ist die Einbindung geeignet ausgewählter und trainierter Neuronaler Netze in eine mehrgliedrige Evolutionsstrategie. Die Neuronalen Netze dienen dabei der Erfassung von Kriterien und Restriktionen eines Vektoroptimierungsproblems. Ziel ist die Untersuchung mehrzieliger Optimierungsaufgaben aus der Strukturmechanik. Die zur simultanen Berücksichtigung mehrerer Optimierungskriterien und -restriktionen entwickelte Systematik ist in Abbildung 20 dargestellt.

Die Komplexität der Lösung einer Tragwerksoptimierungsaufgabe hat ihre Ursache im impliziten Charakter der das Problem beschreibenden Zusammenhänge und Parameter. Die hierzu iterative Vorgehensweise sowohl stochastischer als auch deterministischer Optimierungsverfahren erfordert häufige Neubestimmungen der Strukturreaktion zugehörig zum aktuellen Iterationszustand. Dies geschieht im allgemeinen durch wiederholte FEM-Analysen sowie gekoppelte FEM-Zuverlässigkeitsanalysen. Die entwickelte Systematik ersetzt die aufwendigen und im Rahmen von stochastischen Optimierungsverfahren häufig auftretenden Rechenoperationen (wie z.B. Restriktionskontrollen) für definierte Parameterfenster relevanter Restriktionen und Optimierungskriterien durch trainierte Neuronale Netze.

Der Einsatz trainierter Neuronaler Netze als beschreibende und schnell verfügbare Approximation einzelner Optimierungskriterien bzw. Optimierungsrestriktionen ergänzt in idealer Weise die Natur stochastischer Optimierungsverfahren, da die Systematik stochastischer Optimierungsalgorithmen eine hohe Zeitkomplexität bedingt. Der Einsatz geeigneter Neuronaler Netze zur Approximation von Optimierungskriterien bzw. -restriktionen bietet aufgrund der geringen Antwortzeiten trainierter Neuronaler Netze deutliche Zeitvorteile. Dies liegt in der geringen mathematischen Komplexität bei der Berechnung der zu einem Eingabevektor gehörigen Ausgabe sowie der parallelen Informationsverarbeitung und -speicherung begründet.

Nach der Erläuterung des vorgeschlagenen Grundprinzips zur Erfassung der Optimierungskriterien und -restriktionen einer Vektoroptimierungsaufgabe durch trainierte Neuronale Netze wird die Erfassung der Zuverlässigkeit von Fachwerkstrukturen durch Neuronale Netze besprochen. Basierend auf den Untersuchungen zu Anwendungsmöglichkeiten Neuronaler Netze in der Strukturmechanik in Kapitel 6 folgt die Einbindung der trainierten Neuronalen Netze in die in Abschnitt 3.3 besprochene, mehrgliedrige Evolutionsstrategie. Die durchgeführten Untersuchungen zur Tragwerksoptimierung sind auf die Dimensionsoptimierung statisch be-

stimmter Fachwerke beschränkt. Es gelten die in Abschnitt 6.1.2 getroffenen Einschränkungen. Durch die Untersuchungen soll am Beispiel des Fachwerks FW2 aus Abschnitt 6.2 die grundsätzliche Funktionsfähigkeit der vorgeschlagenen Systematik validiert werden.

7.1 Grundprinzip

Grundsätzlich gibt es zwei verschiedene Möglichkeiten, Neuronale Netze in der Tragwerksoptimierung einzusetzen:

- Training eines direkten Zusammenhangs zwischen verschiedenen Ausgangskonfigurationen der zu untersuchenden Struktur und zugehörigen Optimalstrukturen.
- Training eines Zusammenhangs zwischen charakteristischen Beschreibungsgrößen der Struktur und zugehörigen Strukturantworten wie Knotenverschiebungen, Restriktionen, Sensitivitäten usw. als Reaktion auf Anregungen des Systems. Die trainierten Neuronale Netze dienen dann der Beschreibung von Optimierungskriterien und -restriktionen innerhalb eines Optimierungsalgorithmus.

In Abbildung 20 ist die Grundidee einer mehrzieligen Optimierung durch Einbindung trainierter Neuronaler Netze in Evolutionsstrategien dargestellt. Die Spalten in Abbildung 20 kennzeichnen jeweils das Erfassungs- und Beschreibungsschema eines Optimierungskriteriums bzw. einer Optimierungsrestriktion. Je nach Problemstellung sind diese durch geeignete problemspezifische oder konventionelle Neuronale Netze individuell repräsentiert. Die vorgestellte Systematik erlaubt die parallele Erfassung weiterer Optimierungskriterien bzw. -restriktionen in analoger Weise. Die untersuchte mehrzielige Optimierungsaufgabe ergibt sich aus der simultanen Berücksichtigung mehrerer z.T. multimodaler Optimierungskriterien. Diese sind:

- Minimierung der Versagenswahrscheinlichkeit der Struktur hinsichtlich definierter Versagenskriterien infolge unsicherer Systemeigenschaften. Dies ist der Maximierung der Systemzuverlässigkeit äquivalent.
- Minimierung der Systemmasse

Die Untersuchung der Optimierungsaufgabe berücksichtigt folgende Nebenbedingungen:

- Lastübertragung als zu erfüllende Aufgabe der Statik.

Mehrzieloptimierung durch Evolutionsstrategien unter Einbindung Neuronaler Netze

Anwendungsbeispiel aus der Strukturoptimierung

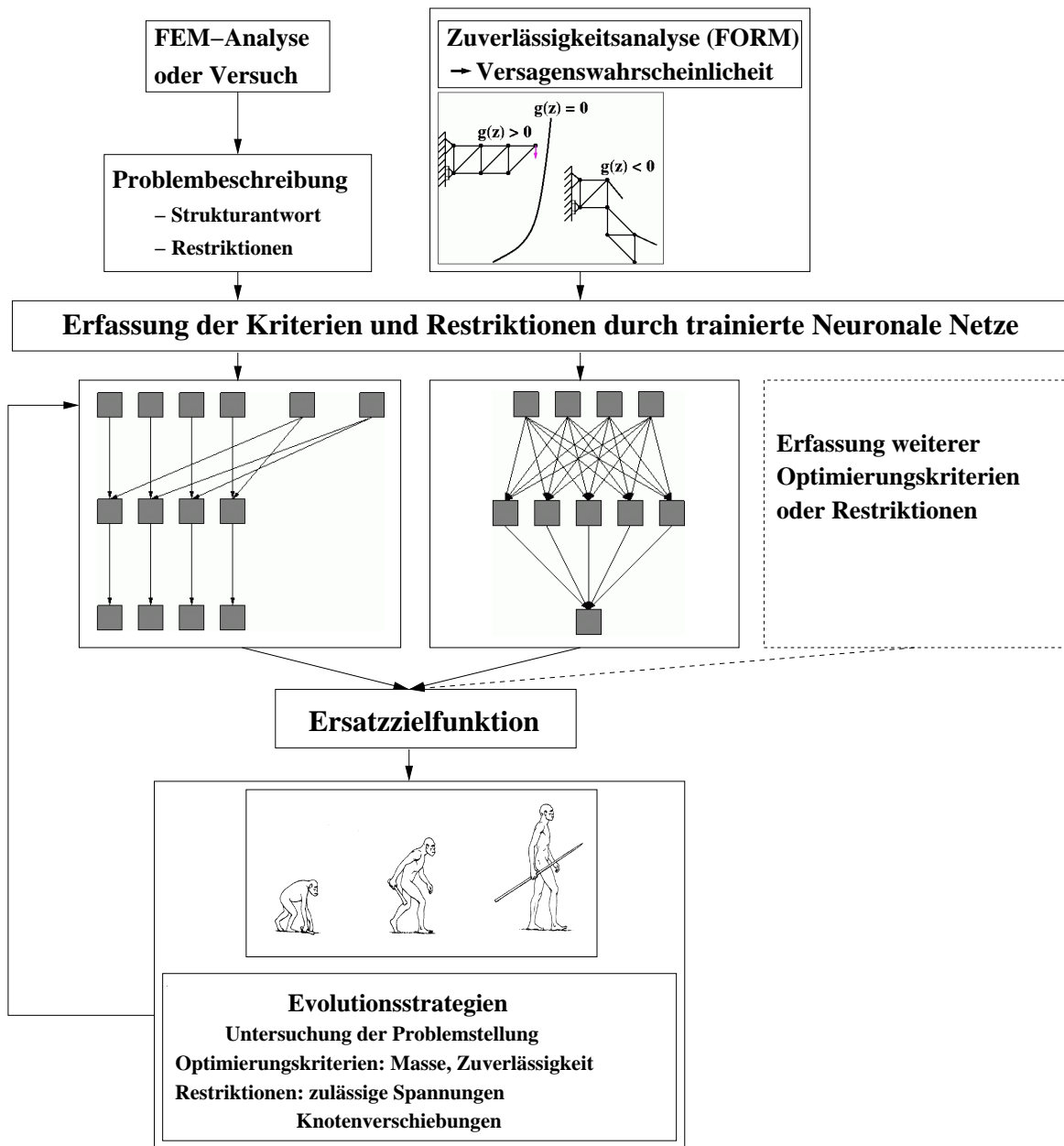
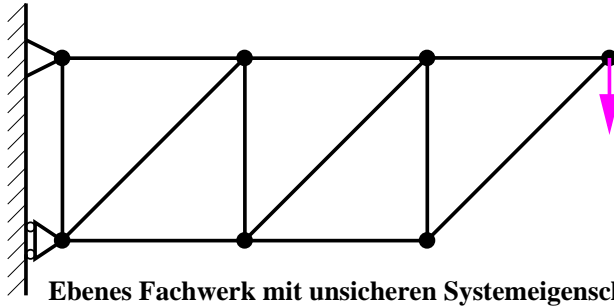


Abbildung 20: Einbindung Neuronaler Netze in Evolutionsstrategien

- Einhaltung maximal zulässiger Stabspannungen (Zug) und maximal zulässiger Knotenverschiebungen

Die simultane Berücksichtigung der Optimierungskriterien und -restriktionen erfolgt über die in Abschnitt 3.2 definierten Ersatzzielfunktionen. Die Stabilität der untersuchten Strukturen wird als ausreichend angenommen. Anwendungsbeispiel für die vorgeschlagene Systematik ist das bereits in Abschnitt 6.2 untersuchte Fachwerk FW2 aus Abbildung 11.

7.2 Beschreibung von Strukturzuverlässigkeiten durch Neuronale Netze

Die Grundlagen der eingesetzten Zuverlässigkeitstheorie sowie die Erzeugung der Trainingsdaten über die Kopplung von Zuverlässigkeits- und FEM-Programm sind in Kapitel 4 beschrieben.

Die Berücksichtigung als unsichere Basisvariable ist grundsätzlich für jeden Parameter des untersuchten Systems möglich. Der in Tabelle 13 definierte Vektor \mathbf{X} der unsicheren Basisvariablen setzt sich im vorliegenden Fall aus den Durchmessern $D_{j,k}$ der kreisförmigen Stäbe des Fachwerks FW2 zwischen den Knoten j und k zusammen. Dies bedeutet, daß die Durchmesser der Stäbe z.B. herstellungsbedingte Schwankungen aufweisen können und Durchmesserangaben mit einer gewissen Unsicherheit bzw. Wahrscheinlichkeit behaftet sind.

Den unsicheren Basisvariablen wird, da negative Durchmesser physikalisch unmöglich sind, eine Lognormalverteilung mit Mittelwert μ und Standardabweichung σ zugeordnet (siehe Abschnitt 4.1). Die Lageparameter erfassen zusammen mit der Wahl der Verteilungsfunktion die z.B. aus dem jeweiligen Herstellungsprozeß resultierenden Unsicherheiten. Zur Erzeugung der Trainingsdaten werden die Stabdurchmesser (Mittelwerte der Lognormalverteilung) durch den Permutationsalgorithmus in individueller Weise variiert. Die Angaben aus Tabelle 13 gelten damit nur exemplarisch. Diese Definition der Basisvariablen ist für die Test- und Trainingsdatensysteme aller Zuverlässigkeitsuntersuchungen dieser Arbeit gültig.

Die Zustandsfunktion g aus Abschnitt 4.3 definiert den untersuchten Versagensfall als Reaktion auf die Belastung der Struktur sowie als Reaktion auf die sicheren und unsicheren Systemeigenschaften. Die Zustandsfunktion g ist aus dem Vergleich einer zulässigen Grenzauslenkung des Knotens 4 in y -Richtung v_{limit} mit der aktuell auftretenden Auslenkung des Knotens 4 in y -Richtung $\delta_{\text{Kn } 4,y}$ zu

$$g(\mathbf{Z}) = \frac{v_{\text{limit}} - \delta_{\text{Kn } 4,y}}{a} \quad (121)$$

mit a als einer charakteristischen Länge der Fachwerks FW2 festgelegt. Die sich aus den

Definition der unsicheren Basisvariablen		
Basisvariablen	\mathbf{X}	$D_{1,2}, D_{2,3}, D_{3,4}, D_{5,6}, D_{6,7},$ $D_{1,7}, D_{2,7}, D_{2,6}, D_{3,6}, D_{3,5}, D_{4,5}$
Dichte der Verteilungsfunktion	$f_{X,\log}(x)$	$\begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} & \text{falls } x > 0 \\ 0 & \text{sonst} \end{cases}$
Mittelwert	μ [mm]	40, 40, 30, 40, 40, 40, 35, 40, 35, 40, 35
Standardabweichung	σ [mm]	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

Tabelle 13: Definition der unsicheren Basisvariablen

Simulationsrechnungen mit FEMZA (siehe Abschnitt 4.5) ergebende Versagenswahrscheinlichkeit P_f ist in den Datensystemen in der Form $\frac{-\ln(P_f)}{10}$ berücksichtigt. Die Transformation sichert die numerische Ausgeglichenheit der auftretenden sehr kleinen Werte von P_f . Die

Test- und Trainingsdatensysteme T_9, T_{10}, T_{11}		
Eingabevektor	\mathbf{in}	$\frac{v_{limit}}{a}, \frac{\delta_{Kn,4,y}}{a}, \Psi$
Sollausgabevektor	\mathbf{o}	$\frac{\ln(P_f)}{-10}$
Trainingsdaten	T_9	$\Psi = 1,2,2,5$ mit 12288 Mustern
Trainingsdaten	T_{10}	$\Psi = 0.8,1,2$ mit 12288 Mustern
Trainingsdaten	T_{11}	$\Psi = 1,2$ mit 8192 Mustern
Testdaten	$T_{9ta}, T_{10ta}, T_{11ta}$	$\Psi = 1.25$ mit 2048 Mustern
Testdaten	$T_{9tb}, T_{10tb}, T_{11tb}$	$\Psi = 1.5$ mit 2048 Mustern
Normierungsintervall]0.05, 0.95[

Tabelle 14: Datensysteme T_9, T_{10}, T_{11} zur Netztopologie NT H

Test- und Trainingsdatensysteme T_9, T_{10}, T_{11} aus Tabelle 14 variieren Kombinationsmöglichkeiten charakteristischer Längenverhältnisse $\Psi = \frac{a}{b}$ des Fachwerks FW2 aus Abbildung 11. Die Neuronen der hierzu entwickelten konventionellen, ebenenweise vollständig vernetzten (3 – 4 – 1)-Netztopologie NT H in Abbildung 21 gehorchen der Transferfunktion aus Gleichung (78) mit $f_{akt,1}(x) = \frac{1}{1+e^{-x}}$. Die durchgeführten Simulationsrechnungen beschränken sich auf die in Tabelle 15 bezeichneten Parametervariationen zu reinem Training der Netztopologien. Die Tabelle 16 zeigt erfolgreiche Ergebnisse zu Test und Training der konven-

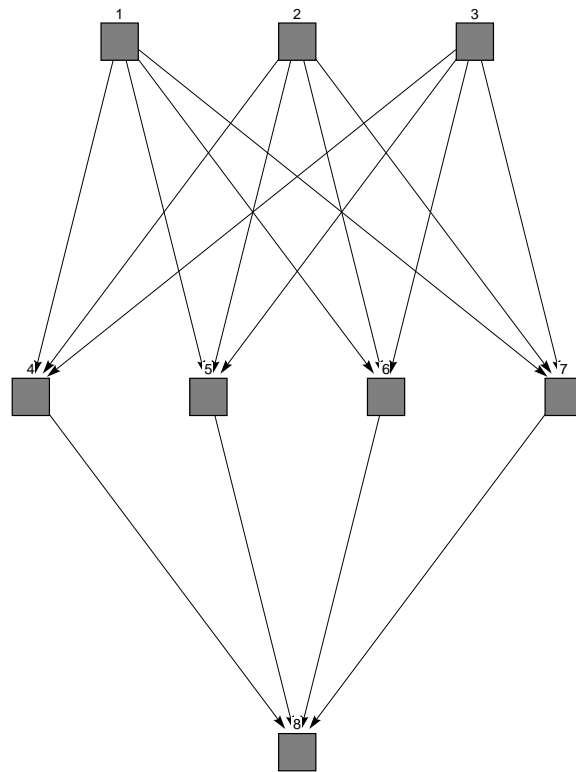


Abbildung 21: Netztopologie NT H

Training NT H			
NT	H	Trainingsalgorithmus	BackpropMomentum
Trainingsdaten	T ₉ , T ₁₀ , T ₁₁	Testdaten	T _{9ta} , T _{10ta} , T _{11ta}
Initialisierungsintervall	[-1.0,1.0]		T _{9tb} , T _{10tb} , T _{11tb}

Tabelle 15: Parameter Training NT H

tionellen Netztopologie NT H. Die Abkürzung „kbE“ kennzeichnet Fälle, bei denen für die untersuchte Parametervariation keine brauchbaren Ergebnisse erzielt werden konnten.

Die Netzübertragungsfunktion der Netztopologie NT H ist durch

$$NF_{NT\ H}(v_{limit}, \delta_{Kn\ 4,y}, \Psi) = f_{akt,n\ 8} \left(\sum_{j=4}^7 \left(w_{j\ 8} f_{akt,n\ j} \left(\sum_{i=1}^3 w_{i\ j} in_i + \theta_j \right) \right) + \theta_8 \right) \quad (122)$$

mit $f_{akt,n\ j} = \left(\frac{1}{1+e^{-x}} \right)$ als sigmoide Aktivierungsfunktion des Neurons j gemäß Gleichung (76) und θ_j als Schwellenwert der Neurons j gegeben. Der Eingabevektor ist gemäß den Festlegungen in Tabelle 14 durch $\mathbf{in} = (v_{limit}, \delta_{Kn\ 4,y}, \Psi)^T$ definiert. Die Versagenswahrscheinlichkeit P_f des Fachwerks FW2 ergibt sich mit Gleichung (122) aus der erfolgreich trainierten

Ergebnisse Training NT H				
NT	Daten		Lernalgorithmus	Test in [%]
	Test	Training		μ_{test}
NT H	T _{9ta}	T ₉	BackpropMomentum	12.99
NT H	T _{9tb}	T ₉	BackpropMomentum	5.11
NT H	T _{10ta}	T ₁₀	BackpropMomentum	kbE
NT H	T _{10tb}	T ₁₀	BackpropMomentum	kbE
NT H	T _{11ta}	T ₁₁	BackpropMomentum	kbE
NT H	T _{11tb}	T ₁₁	BackpropMomentum	18.05

Tabelle 16: Ergebnisse Training NT H

Netztopologie NT H in expliziter Form zu

$$P_f = e^{-10 f_{\text{akt},n} \left(\sum_{j=4}^7 \left(w_{j,8} f_{\text{akt},n,j} \left(\sum_{i=1}^3 w_{i,j} \text{in}_i + \theta_j \right) \right) \right) + \theta_8} . \quad (123)$$

Die Matrix der Verbindungsgewichte \mathbf{W} und der Vektor der Schwellenwerte $\bar{\theta}$ nehmen folgende Werte an:

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & 0 & 15.99993 & -5.39850 & 0.23000 & 5.68693 & 0 \\ 0 & 0 & 0 & -28.74750 & 7.57652 & 31.99999 & -6.08137 & 0 \\ 0 & 0 & 0 & -3.18638 & -0.50000 & -1.30006 & 0.28235 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15.99998 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -14.11585 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -8.00019 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.67582 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (124)$$

$$\bar{\theta} = (\theta_4, \dots, \theta_8)^T = (1.62296, -7.71522, 0.97826, -0.47256, 2.85489)^T .$$

Für die Neuronen der Eingabeschicht werden keine Aktivierungsfunktionen und damit auch keine Schwellenwerte berücksichtigt (siehe Abschnitt 5.2.2). Die in den durchgeführten Simulationsrechnungen erzielten Ergebnisse sind als befriedigend zu bezeichnen. Grundsätzlich ist hiermit die Erfassbarkeit von Informationen zum Zuverlässigkeitsverhalten der untersuchten Fachwerkstruktur nachgewiesen. Weitere untersuchte konventionelle Netztopologien wie z.B. (14 – 15 – 15 – 1)-Netze mit Eingabevektor $\mathbf{in} = \left(\Pi_1, \dots, \Pi_{11}, \frac{1}{\Psi^3}, \left(1 + \frac{1}{\Psi^2} \right)^{\frac{3}{2}}, \frac{v_{\text{limit}}}{a} \right)^T$ und Ausgabevektor $\mathbf{o} = \left(\frac{\ln(P_f)}{-10} \right)$ zeigen nur unzureichende Trainings- bzw. Testergebnisse.

In einem Ausblick zu dieser Teilthematik soll auf die Möglichkeit hingewiesen werden, ebenenweise vollständig vernetzte konventionelle Ausgangsnetztopologien durch *Pruning* zu vereinfachen. Untersuchungen zum Vorhersagepotential von Versagenswahrscheinlichkeiten ausgedünnter Netztopologien relativ zu reinen Trainingsergebnissen stehen noch aus. Auch könnten geeignete problemspezifische Ausgangsnetztopologien mit eingebautem Funktionsansatz - z.B. Polynomansätze höherer Ordnung - das Zuverlässigkeitsverhalten hinreichend erfassen. In Anlehnung an die Datensysteme T_9, T_{10}, T_{11} sei hier z.B. der Polynomansatz

$$NF_1(x_1, x_2, x_3) = [x_1 + x_1^2 + x_1^3] [x_2 + x_2^2 + x_2^3] [x_3 + x_3^2 + x_3^3] \quad (125)$$

mit $x_1 = \frac{v_{limit}}{a}$, $x_2 = \frac{\delta_{Kn, 4,y}}{a}$, $x_3 = \Psi$ genannt.

7.3 Einbindung von Neuronalen Netzen in Evolutionsstrategien

Für die in Abbildung 20 vorgeschlagene und vom Netzsimulator unabhängige Grundsystematik der Anbindung erfolgreich trainierter Neuronaler Netze an ein Optimierungsverfahren bestehen verschiedene Möglichkeiten:

- Äquivalente Umsetzung der Netzübertragungsfunktion in C Code, der als C-Funktion aufgerufen werden kann. Dies wird durch SNNS [104] unterstützt und bietet sich für komplexe konventionelle Netztopologien an.
- Abstraktion der Netzübertragungsfunktion in Form eines mathematischen Zusammenhangs. Dies bietet sich insbesondere für problemspezifische und nichtkomplexe, konventionelle Netztopologien an und wird hier eingesetzt.

Der in Abbildung 20 dargestellte Rücksprung aus dem Optimierungsalgorithmus in die Problembeschreibungsebene symbolisiert die Abfrage neuer Systemreaktionen und Restriktionsaussagen durch den Optimierungsalgorithmus aus den trainierten Neuronalen Netzen zugehörig zum aktuellen Iterationszustand. Die Wertebereiche der so aus trainierten Neuronalen Netzen abrufbaren Parameterkombinationen müssen a priori abgeschätzt werden, um die Trainingsdatensysteme entsprechend auslegen zu können. Die Abdeckung eines breiten Wertebereichs an Systemreaktionen hinsichtlich zu erfassender Optimierungskriterien und -restriktionen ist für konventionelle Neuronale Netze durch das Training mehrerer, sich in den nebeneinanderliegenden Wertebereichen ihrer Datensysteme ergänzender Netze realisierbar.

Die Zuordnung der aktuell angefragten Systemreaktionen (Optimierungskriterien und -restriktionen) zu den einzelnen Bereichsnetzen kann über ein Expertensystem oder ein entsprechend trainiertes Neuronales Netz erfolgen. Auf diese Weise ist ein großer Wertebereich

an Systemantworten durch mehrere gleichartige, auf schmale Einzelbereiche spezialisierte konventionelle Netztopologien erfassbar.

Problemspezifische Neuronale Netze sind dagegen in der Lage, den zu beschreibenden Zusammenhang analytisch exakt über dem gesamten Definitionsbereich des untersuchten Zusammenhangs zu erfassen (siehe Abschnitt 6.1).

Der Rückfall auf eine herkömmliche Berechnung angefragter Systemreaktionen z.B. durch FEM-Analyse ist für jene Parameterkombinationen vorgesehen, die aufgrund des stochastischen Charakters des Optimierungsverfahrens außerhalb der durch trainierte Neuronale Netze erfaßten Parameterintervalle liegen.

7.3.1 Berücksichtigung der Optimierungskriterien

Die Berücksichtigung der aktuellen Masse des Fachwerks erfolgt über eine explizite mathematische Formulierung als Teil der Ersatzzielfunktion. Die Länge $l_{j k}$ des Stabs zwischen den Knoten j und k eines ebenen, unbelasteten Fachwerks ist durch den euklidischen Abstand der Knoten j und k mit (x_j, y_j) als kartesischen Koordinaten des Knotens j zu

$$l_{j k} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (126)$$

gegeben. Die Systemmasse M_{FW} ergibt sich aus der aktuellen Geometrie des Fachwerks und der Materialdichte $\rho = 2.8 \cdot 10^{-6} \frac{\text{kg}}{\text{mm}^3}$ der eingesetzten Al-Legierung 3.4354 (Werkstoff-Leistungsblatt WL 3.4354 bzw. Luftfahrttechnisches Handbuch [128]) zu

$$M_{FW} = \rho \sum_{\text{Stäbe}} l_{j k} \pi \frac{D_{j k}^2}{4}. \quad (127)$$

Die Erfassung der Versagenswahrscheinlichkeit P_f erfolgt über die in Abschnitt 7.2 beschriebene Systematik.

7.3.2 Berücksichtigung der Optimierungsrestriktionen

Die eingebundenen Neuronale Netze zur Beschreibung der Optimierungsrestriktionen Knotenverschiebungen und Stabspannungen basieren auf den in Kapitel 6 untersuchten Anwendungsmöglichkeiten Neuronaler Netze in der Strukturmechanik. Die Einbindung der eingesetzten problemspezifischen Neuronale Netze in den Optimierungsalgorithmus erfolgt als expliziter mathematischer Zusammenhang, aufgebaut aus den erfolgreich trainierten und u.U. durch *Pruning* reduzierten, problemspezifischen Ausgangsnetztopologien.

7.3.3 Ersatzzielfunktion

Die Erfassung und Skalarisierung der mehrzieligen Optimierungsaufgabe aus Gleichung (42) erfolgt über die in Abschnitt 3.2 in Gleichung (41) spezifizierte Ersatzzielfunktion \mathbf{Ez}_3 als gewichtete Summierung der Einzelzielfunktionen f_j .

7.4 Mehrzieloptimierung des Fachwerks FW2

Die Untersuchung einer mehrzieligen Optimierungsaufgabe über eine Ersatzformulierung dient der Validierung der in Abbildung 20 vorgeschlagenen Systematik. Anwendungsbeispiel ist die Minimierung von Masse und Versagenswahrscheinlichkeit des Fachwerks FW2 unter Berücksichtigung gegebener Restriktionen. Die Erfassung der Optimierungskriterien erfolgt gemäß den Festlegungen des Abschnitts 7.3.1. Die Erfassung der Optimierungsrestriktionen (maximal zulässige Knotenverschiebungen und maximal zulässige Stabspannungen) erfolgt über erfolgreich trainierte und durch *Pruning* reduzierte, problemspezifische Neuronale Netze gemäß den Ergebnissen der Abschnitte 6.4 und 6.5. Die Anbindung der trainierten Neuronalen Netze an die in Abschnitt 3.3 besprochenem, mehrgliedrige Evolutionsstrategie erfolgt gemäß Abschnitt 7.3 in Form expliziter mathematischer Funktionen.

7.4.1 Ersatzzielfunktion und Optimierungsrestriktionen

Die eingesetzte Ersatzzielfunktion \mathbf{Ez}_3 setzt sich aus der Einzelzielfunktion f_1 zur Minimierung der Masse M_{FW2} des Fachwerks FW2 gemäß Gleichung (127) und f_2 zur Minimierung des natürlichen Logarithmus' der Versagenswahrscheinlichkeit $\ln(P_{f FW2})$ des Fachwerks FW2 mit $P_{f FW2}$ gemäß Gleichung (123) – jeweils in normierter Form – zusammen. Die Einzelzielfunktionen sind durch

$$f_1 = \frac{M_{FW2}}{M_{FW2 \text{ norm}}} = \rho \sum_{\text{Stäbe}} \left(l_{j k} \pi \frac{D_{j k}^2}{4} \right) \frac{1}{M_{FW2 \text{ norm}}} \quad (128)$$

$$f_2 = \frac{\ln(P_{f FW2 \text{ norm}})}{\ln(P_{f FW2})} = \frac{\ln(P_{f FW2 \text{ norm}})}{-10 f_{\text{akt},n 8} \left(\sum_{j=4}^7 \left(w_{j 8} f_{\text{akt},n j} \left(\sum_{i=1}^3 w_{i j} \text{in}_i + \theta_j \right) \right) \right) + \theta_8}$$

gegeben. Die Versagenswahrscheinlichkeit P_f wird aufgrund ihres mehrere Größenordnungen umfassenden Wertebereichs durch den natürlichen Logarithmus \ln auf numerisch ausgeglichene Werte transformiert. Die Transformation ändert durch den monotonen Verlauf des natürlichen Logarithmus nichts an der qualitativen Aussage des Optimierungskriteriums. Gleichzeitig zeigt sich aber umgekehrt proportionales Verhalten zwischen der Versagenswahrschein-

lichkeit P_f bzw. $\ln(P_f)$ und einem auf positive Werte normierten Versagensindex $\frac{\ln(P_f \text{ FW2})}{\ln(P_f \text{ FW2 norm})}$. Die angestrebte Minimierung des auf positive Werte normierten Versagensindex' wäre dann jedoch mit ansteigenden Werten für P_f verbunden. Um der gegebenen Minimierungsaufgabe gerecht zu werden, wird stattdessen die in Gleichung (128) dargestellte, inverse Anordnung des Versagensindex' für das Optimierungskriterium f_2 verwendet.

Die Normierungen der Einzelzielfunktionen f_1 bzw. f_2 durch die Masse FW2 $M_{\text{FW2 norm}}$ bzw. den natürlichen Logarithmus der Versagenswahrscheinlichkeit $\ln(P_f \text{ FW2 norm})$ einer Referenztopologie des Fachwerks FW2 sichert ihre numerische Ausgeglichenheit und Vergleichbarkeit. Es gelte die nachfolgende Gewichtung der normierten Einzelzielfunktionen zur Ersatzzielfunktion

$$\mathbf{Ez}_{\text{FW2}} = \alpha_1 f_1 + \alpha_2 f_2 = \min! . \quad (129)$$

Die Faktoren α_i mit $0 \leq \alpha_i \leq 100$ und $\alpha_1 + \alpha_2 = 100$ ermöglichen eine variable, relative Wichtung der Einzelzielfunktionen zueinander und garantieren numerisch ausgeglichene Werte der Ersatzzielfunktion \mathbf{Ez}_{FW2} .

Die in den Test- und Trainingsdatensystemen zur Erfassung der Versagenswahrscheinlichkeit des Fachwerks FW2 (z.B. T_9) vorgenommene Normierung zugehörig zur eingesetzten Neuronenfunktionalität (siehe Abschnitt 5.2.2) wird bei der Einbindung der Netzübertragungsfunktion der Netztopologie NT H in den Optimierungsalgorithmus über Rücktransformationen berücksichtigt. Die Optimierungsrestriktion $g_i \leq \text{limit}_{\delta_{\text{Kn } i,y}}$ zur Begrenzung zulässiger Knotenverschiebungen $\delta_{\text{Kn } i,y}$ der Knoten $i = 1, \dots, 7$ des Fachwerks FW2 und $g_{j,k} \leq \text{limit}_{\sigma_{j,k}}$ zur Begrenzung zulässiger Stabspannungen $\sigma_{j,k}$ des Stabs $\text{Stab}_{j,k}$ zwischen den Knoten j und k des Fachwerks FW2 durch erfolgreich trainierte Neuronale Netze entsprechen den Ergebnissen der Abschnitte 6.4 bzw. 6.5. Als Beispiel zur Erfassung aller Verschiebungen $\delta_{\text{Kn } i,y}$ der Knoten des Fachwerks FW2 in y -Richtung sei das Netzwerk NT okt53o in Abbildung 17 bzw. die zugehörigen Netzübertragungsfunktionen genannt. Als Beispiel zur Erfassung aller Normalspannungen der Stäbe $\text{Stab}_{j,k}$ sei die aus den Ausgangsnetztopologien ANT F und ANT G durch *Pruning* ermittelte Netztopologie NT D in Abbildung 15 bzw. die zugehörigen Netzübertragungsfunktionen genannt.

Durch die Festlegung der Trainingsdaten T_9 aus dem Permutationsalgorithmus durch $\Psi = 1$ mit $v_{\text{limit}} = 4.6 \cdot 10^{-2}$, $\Psi = 2$ mit $v_{\text{limit}} = 1.5 \cdot 10^{-2}$, $\Psi = 2.5$ mit $v_{\text{limit}} = 1.3 \cdot 10^{-2}$ hinsichtlich der Geometriedaten und durch

Stab _{j,k}	D _{1,2}	D _{2,3}	D _{3,4}	D _{5,6}	D _{6,7}	D _{1,7}	D _{2,7}	D _{2,6}	D _{3,6}	D _{3,5}	D _{4,5}
D [mm]	40,42	40,42	30,32	40,42	40,42	40,42	35,37	40,42	35,37	40,42	35,37

hinsichtlich der erfaßten Intervalle der Stabdurchmesser ist der Gültigkeitsbereich der Netzübertragungsfunktion $NF_{NT\ H}$ des erfolgreich trainierten Netzwerks $NT\ H$ aus Abschnitt 7.2 gegeben. Die untersuchten Knotenlasten sind $P = 1.0 \cdot 10^5\ N$ sowie $P = 1.2 \cdot 10^5\ N$. Der schmale Bereich erfaßter Intervalle der Stabdurchmesser kann durch sich ergänzende Netzwerke gemäß der in Abschnitt 7.3 vorgeschlagenen Methode erweitert werden.

7.4.2 Optimierung des Fachwerks FW2 mit (10,100)-Evolutionsstrategie

Die Festlegung der Strategieparameter der mehrgliedrigen Evolutionsstrategie erfolgt nach Empfehlungen von SCHWEFEL [77] und Erfahrungen des Autors in STRENG [76]. Die Strategieparameter

μ	λ	$lparen$	$loldie$	$maxzyk$	$maxage$	$maxgen$	$maxpop$	$migras$
10	100	10	0	1	0	1000	1	0

definieren einen Evolutionslauf der (10,100)-Evolutionsstrategie mit einer isolierten Population, die für einen Zyklus von 1000 Generationen die Optimierungsaufgabe in Gleichung (129) untersucht. Zur Simulation der gewählten Evolutionsstrategie wird eine erweiterte Form des Programms KORR nach SCHWEFEL [77] eingesetzt.

Der Vektor der Objektparameter eines Individuums ist durch

$$\mathbf{x} = (D_{1\ 2}, D_{2\ 3}, D_{3\ 4}, D_{5\ 6}, D_{6\ 7}, D_{1\ 7}, D_{2\ 7}, D_{2\ 6}, D_{3\ 6}, D_{3\ 5}, D_{4\ 5})^T \quad (130)$$

gegeben. Restriktionen sind maximal zulässige Normalspannungen $\text{limit}_{\sigma_j\ k}$ der Stäbe $Stab_j\ k$, eine maximal zulässige Verschiebung des Knotens 4 $\text{limit}_{\delta_{Kn\ 4,y}}$ in y-Richtung sowie die sich aus den Trainingsdaten T_9 ergebenden, zulässigen Wertebereiche der in $NT\ H$ erfaßten Stabdurchmesser. Als Abbruchkriterium wird die Differenz der Mittelwerte der Ersatzzielfunktionswerte MO der Objektvariablen aller Individuen der untersuchten Population nach einer festgelegten Anzahl aufeinanderfolgender Generationen verglichen. Das Abbruchkriterium beendet den Evolutionsalgorithmus, falls MO den absoluten Grenzwert $\epsilon_A = 10^{-7}$ oder den relativen Grenzwert $\epsilon_R = 10^{-7}$ erreicht bzw. unterschreitet. Dies vermeidet unnötige Rechnungen bei sehr kleinen Änderungen von MO . Grund könnte dabei die Lokalisierung eines lokalen oder des gesuchten globalen Minimums der Ersatzzielfunktion sein.

Die folgende Tabelle führt Startwerte (Start) und Ergebnisse verschiedener Evolutionsläufe (SIMi) nach erzielter Versagenswahrscheinlichkeit P_f und erzielten Stabdurchmessern $D_j\ k$ [mm] auf:

Daten	P_f	$D_{1\ 2}$	$D_{2\ 3}$	$D_{3\ 4}$	$D_{5\ 6}$	$D_{6\ 7}$	$D_{1\ 7}$	$D_{2\ 7}$	$D_{2\ 6}$	$D_{3\ 6}$	$D_{3\ 5}$	$D_{4\ 5}$
Start	$6.62 \cdot 10^{-14}$	42	42	32	42	42	42	37	42	37	42	37
SIM5	$2.81 \cdot 10^{-12}$	42	42	32	40	42	40	37	40	37	40	37
SIM10	$6.54 \cdot 10^{-10}$	42	42	32	40	40	40	35	40	37	40	35

Die Wichtungsfaktoren α_i und die erzielte relative Reduzierung der Fachwerksmasse $\Delta M_{FW2} = \frac{M_{FW2\ Start} - M_{FW2}}{M_{FW2\ Start}}$ haben für die dargestellten Simulationsergebnisse folgende Werte:

Daten	α_1	α_2	ΔM_{FW2}
SIM5	90	10	3.42 %
SIM10	95	5	6.40 %

Die Simulationsergebnisse $D_{j\ k}$ und P_f zu SIM5 und SIM10 stellen lokale Optima im Sinne der Optimierungsaufgabe und beschränkt durch die gegebenen Restriktionen dar.

Durchgeführte Simulationsrechnungen zu umgekehrten Verhältnissen der α_i oder zu $\alpha_1 = \alpha_2 = 50$ stagnieren in der Starttopologie des Fachwerks. Es wird deutlich, daß die Reduktion der Stabquerschnittsflächen und damit die gewünschte Reduzierung der Fachwerksmasse gegen eine Erhöhung der Versagenswahrscheinlichkeit eingetauscht werden muß (siehe obige Tabelle). Grundsätzlich ist damit die Funktionsfähigkeit der vorgestellten Methode demonstriert.

7.5 Fazit

Die entwickelte Methode beruht auf neuen Untersuchungen zur Einbindung des Abstraktions- und Lernpotentials Neuronaler Netze in stochastische Optimierungsverfahren wie mehrgliedrige Evolutionsstrategien.

Die Struktur dieser Methode ist in Abbildung 20 dargestellt und nutzt die Synergiepotentiale beider Methoden zur Untersuchung mehrzieliger Optimierungsaufgaben. Dies betrifft den Einsatz problemspezifischer Neuronaler Netze, die Erfassung von Optimierungskriterien und -restriktionen durch trainierte bzw. durch ausgedünnte Neuronale Netze (*Pruning*), den Einsatz des stochastischen Optimierungsverfahrens mehrgliedrige Evolutionsstrategie sowie die Möglichkeit, mehrere Optimierungskriterien und -restriktionen parallel zu erfassen. Der Restfehler der eingesetzten konventionellen Neuronale Netze insbesondere bei der Erfassung des

Optimierungskriteriums Versagenswahrscheinlichkeit wirkt sich dagegen negativ auf den erzielbaren Optimierungsfortschritt aus. Die Untersuchungsergebnisse aus Kapitel 6 sind hierzu grundlegend.

Zugehörig zur aktuellen Optimierungssituation können die Reaktionen des untersuchten Tragwerks hinsichtlich der Optimierungskriterien und Restriktionen mit geringem Rechenaufwand ermittelt und über die Einzelzielfunktionen sowie die Ersatzzielfunktion der aktuelle Zielfunktionswert dem Optimierungsverfahren zur Verfügung gestellt werden. Anwendungsbeispiel ist eine Optimierungsaufgabe, die mehrere multimodale Einzelzielfunktionen gleichzeitig berücksichtigt.

Mit steigender Problemkomplexität nimmt der Zeit- bzw. Rechenaufwand für die herkömmliche Ermittlung von Systemreaktionen sowie Optimierungsrestriktionen zur Bestimmung von Einzelzielfunktionen der Ersatzzielfunktion zu. Bei der Erfassung von Kriterien und Restriktionen einer Optimierungsaufgabe durch trainierte Neuronale Netze nimmt mit steigender Problemkomplexität und -größe der Aufwand zur Erzeugung der Trainingsdatensysteme sowie der Aufwand zum Training bzw. *Pruning* der Netzwerke ebenfalls zu. Gleichzeitig zeigt sich aber eine mit der Iterationszahl des Optimierungsverfahrens zunehmende Zeitersparnis durch die geringen Antwortzeiten trainierter Neuronaler Netze. Diese signifikante Steigerung der Effizienz gewinnt besonders in Verbindung mit der großen Anzahl notwendiger Iterationsschritte bei stochastischen Optimierungsverfahren stark an Bedeutung.

Die vorgestellte Methode zur Einbindung des Abstraktions- und Lernpotentials Neuronaler Netze in stochastische Optimierungsverfahren ist nicht auf den Bereich der Tragwerksoptimierung beschränkt, sondern vielseitig anwendbar. Die bisher gewonnenen Erkenntnisse insbesondere zum Beschreibungspotential Neuronaler Netze und den Möglichkeiten zur Ausdünnung Neuronaler Netze werden im folgenden Kapitel auf die Analyse von Schädigungen ebener Fachwerkstrukturen übertragen.

8 Schadensanalyse in Fachwerkstrukturen

Ziel der Untersuchungen zur Schadensanalyse in Fachwerkstrukturen ist es, aus einfach meßbaren Größen einer statisch belasteten Struktur (z.B. Systemreaktionen) Rückschlüsse auf den Schädigungszustand der Struktur nach Existenz, Ort und Ausmaß der Schädigungen ziehen zu können. Dies ist mathematisch der Bestimmung einer unbekanntenen inversen Funktion äquivalent.

Für den Anwendungsfall Schadensanalyse in Fachwerkstrukturen wird demonstriert, wie ohne Kenntnisse und Informationen zur Problemstellung die Approximation eines gesuchten inversen Zusammenhangs möglich ist. Das einer Schadensanalyse dabei zugrundeliegende Mustererkennungsproblem wird im folgenden Ansatz durch Neuronale Netze untersucht. Bei der vorgeschlagenen neuen Methode werden vollständig ebenenweise vernetzte, konventionelle Neuronale Netze zur Erfassung des gesuchten inversen Zusammenhangs trainiert sowie durch das *Pruning*-Verfahren OBS z.T. signifikant im Umfang hinsichtlich überflüssiger verdeckter Neuronen und *links* im Netzplan reduziert.

Der erste Abschnitt dieses Kapitels beschreibt mathematische Grundlagen der Bestimmung inverser Zusammenhänge. Es folgt die Vorstellung einer Systematik zur Erfassung inverser Zusammenhänge mittels Neuronaler Netze, ihrer Limitierungen, zugehöriger Test- und Trainingsdaten, zugehöriger Netztopologien sowie deren Reduktion durch den *Pruning*-Algorithmus OBS. Die vorgeschlagene Methodik wird anschließend am Beispiel statisch bestimmter und unbestimmter Fachwerke untersucht.

8.1 Das inverse Problem

Die zu einer eindeutigen Abbildung f mit Definitionsbereich $\mathbb{D}(f) \subseteq \mathbb{R}$ und Wertebereich $\mathbb{W}(f) \subseteq \mathbb{R}$ zugehörige inverse Abbildung f^{-1} heißt Umkehrfunktion oder inverse Funktion von f . Für die zugehörigen Definitions- und Wertebereiche gilt:

$$\begin{aligned} \mathbb{D}(f^{-1}) &= \mathbb{W}(f) \quad \text{bzw.} \quad \mathbb{W}(f^{-1}) = \mathbb{D}(f) \\ f[f^{-1}(x)] &= x \quad \text{mit } x \in \mathbb{D}(f^{-1}) \quad \text{bzw.} \quad f^{-1}[f(x)] = x \quad \text{mit } x \in \mathbb{D}(f) \\ y = f(x) &\Leftrightarrow x = f^{-1}(y) . \end{aligned} \quad (131)$$

Die Abbildung $f: \mathbb{M}_1 \rightarrow \mathbb{M}_2$ zwischen den Mengen \mathbb{M}_1 und \mathbb{M}_2 heißt Injektion, falls f eindeutig ist. Die Abbildung $f: \mathbb{M}_1 \rightarrow \mathbb{M}_2$ heißt Surjektion, falls $\mathbb{W}(f) = \mathbb{M}_2$ ist. Die Abbildung $f: \mathbb{M}_1 \rightarrow \mathbb{M}_2$ heißt Bijektion, falls $f: \mathbb{M}_1 \rightarrow \mathbb{M}_2$ Injektion und Surjektion ist. Bijektion ist

hinreichend für die eindeutige Bestimmbarkeit eines inversen Zusammenhangs. Es existieren drei Lösungsmöglichkeiten inverser Probleme:

- Direkte Methode: Die gesuchte Parameterverteilung kann direkt aus einer über dem gesamten Gebiet bekannten, das System beschreibenden Größe berechnet werden.
- Indirekte Methode: Die gesuchte Parameterverteilung wird aus einem Vergleich gemessener Größen und den Ergebnissen einer gegebenen Parameterverteilung ermittelt.
- Nichtinvasive Methode: Eine Anregung auf dem Rand des untersuchten Gebiets wirkt sich über die im Inneren des Systems vorhandene Parameterverteilung auf meßbare Werte am Rand des Systems aus.

Übertragen auf die Schadensanalyse von Fachwerkstrukturen erfassen, zugehörig zum aktuellen Zustand der Struktur, in der vorwärts gerichteten Analyse die Funktionen f und g die sich als Resultat einer aufgebrachten Anregung (Ursache) ergebenden Systemreaktionen (Wirkung) der Struktur. Für die Analyse unbekannter Schädigungszustände einer Struktur aus Systemreaktionen der angeregten Struktur wird aber die zugehörige, rückwärtsgerichtete Abbildung benötigt.

Die verschiedenen Schädigungszustände der Fachwerkstruktur sind durch die Reststeifigkeit G der geschädigten Stäbe definiert. Es existiert ein expliziter Zusammenhang f , der den statischen Lastfall F und die Reststeifigkeiten G der Stäbe auf die Knotenverschiebungen \mathbf{u} mit

$$\mathbf{u} = f(G, F) \quad \sigma = g(G, F) \quad (132)$$

des Fachwerks abbildet. Die Normalspannungen σ der Stäbe als explizite Funktion g des Lastfalls F und der Reststeifigkeiten G folgen aus einer Nachlaufrechnung. Dies wird als vorwärtsgerichtete Abbildung expliziter Zusammenhänge angesehen.

Die für die Bestimmung von f^{-1} und g^{-1} durch Neuronale Netze benötigten kausal zusammenhängenden Datenmuster der Vektorpaare des Ursache-Wirkungs-Zusammenhangs \mathbf{u} zugehörig zu (G, F) und σ zugehörig zu (G, F) können experimentell oder über FEM-Simulationen bestimmt werden (siehe Abschnitt 8.2.3). Die Abbildungen f und g sind implizit durch die kausal zusammenhängenden Vektorpaare der das Problem beschreibenden Datenmuster der Test- und Trainingsdaten gegeben. Die Bestimmung der gesuchten inversen Funktionen f^{-1} bzw. g^{-1} mit

$$G = f^{-1}(\mathbf{u}, F) \quad G = g^{-1}(\sigma, F) \quad (133)$$

durch Neuronale Netze macht die Umkehrung der Ursache-Wirkungs-Zusammenhänge notwendig. Der Vektor der Knotenverschiebungen des Fachwerks ist eine äußere und damit auf dem Rand des Systems meßbare Größe. Die als zusätzliche Systemreaktionen berücksichtigten Normalspannungen σ der Stäbe sind Größen aus dem Inneren des Systems. Die Bestimmung der inversen Funktionen erfolgt damit für f^{-1} durch die nichtinvasive Methode und für g^{-1} durch die direkte Methode.

Das Grundprinzip der Schadensanalyse durch Neuronale Netze ist schematisch in Abbildung 22 dargestellt und in Abschnitt 8.2.1 besprochen. Eine ausreichende Verteilung der Trainingsdaten zusammen mit der Eindeutigkeit der Merkmalsarten sind notwendige Voraussetzungen, um den gesuchten inversen Zusammenhang der Schadensanalyse eindeutig zu bestimmen und Mehrdeutigkeiten zu vermeiden. Eine theoretische Diskussion der Zusammenhänge findet sich bei SYLVESTER UND UHLMANN [26]. Einen umfassenden Überblick zum Themenkomplex inverser Probleme gibt z.B. LOUIS [129].

8.2 Schadensanalyse mit Neuronalen Netzen

Im folgenden wird das Grundprinzip zur Bestimmung von Existenz, Ort und Ausmaß eines oder mehrerer gleichzeitig auftretender Schadenszustände einer Struktur durch trainierte Neuronale Netze vorgestellt. Eine direkte Berechnung von Schadenszuständen aus den Systemreaktionen der belasteten Struktur ist mit herkömmlichen FE-Methoden nicht möglich. In Erweiterung der Überlegungen von HERRMANN [10] (siehe Abschnitt 1.2.3) werden zusätzlich statisch unbestimmte Fachwerkstrukturen sowie die Reduktion ebenenweise vollständig vernetzter konventioneller Neuronaler Netze durch das *Pruning*-Verfahren OBS untersucht.

8.2.1 Systematik

Neuronale Netze erfassen, beschränkt durch die Limitierungen der Übertragungsfunktion der Ausgangsnetztopologie und den Lernerfolg des Trainingsalgorithmus', den zwischen Eingabe und Sollausgabe der Trainingsdaten beschriebenen Zusammenhang. Dieser Aspekt wird durch Umkehrung von Ursache und Wirkung vorhandener Informationen über die Problemstellung zur Bestimmung des gesuchten inversen Zusammenhangs genutzt. Der Algorithmus zur Untersuchung geschädigter Strukturen durch Neuronale Netze hat folgenden Aufbau:

- Bestimmung von Systemreaktionen (Stabspannungen σ , Knotenverschiebungen \mathbf{u}) als Wirkung zugehörig zu den Ursachen „geschädigte bzw. ungeschädigte Struktur“ und „äußere statische Belastung“

- Die Umkehrung der Ursache-Wirkungs-Zusammenhänge liefert den benötigten Wirkungs-Ursache-Zusammenhang. Die Datenmuster sind Vektorpaare, die implizit den zur Schadensanalyse benötigten, inversen Zusammenhang beschreiben.
- Training bzw. *Pruning* mit Nachtraining der Ausgangsnetztopologien mit normierten Wirkungs-Ursache-Datenmustern
- Test der trainierten bzw. ausgedünnten Netzwerke mit Testdaten

Ein erfolgreich trainiertes bzw. durch *Pruning* reduziertes Netz ist in der Lage, für den Wertebereich der Trainingsmuster Aussagen zu Existenz, Ort und Umfang einer einzelnen oder mehrerer gleichzeitig auftretender Schädigungen der Struktur zu treffen. Das Grundprinzip der Schadensanalyse statisch belasteter Strukturen durch Neuronale Netze ist in Abbildung 22 dargestellt. Dem Netzwerk fällt die Aufgabe zu, die für die Schadensanalyse rele-

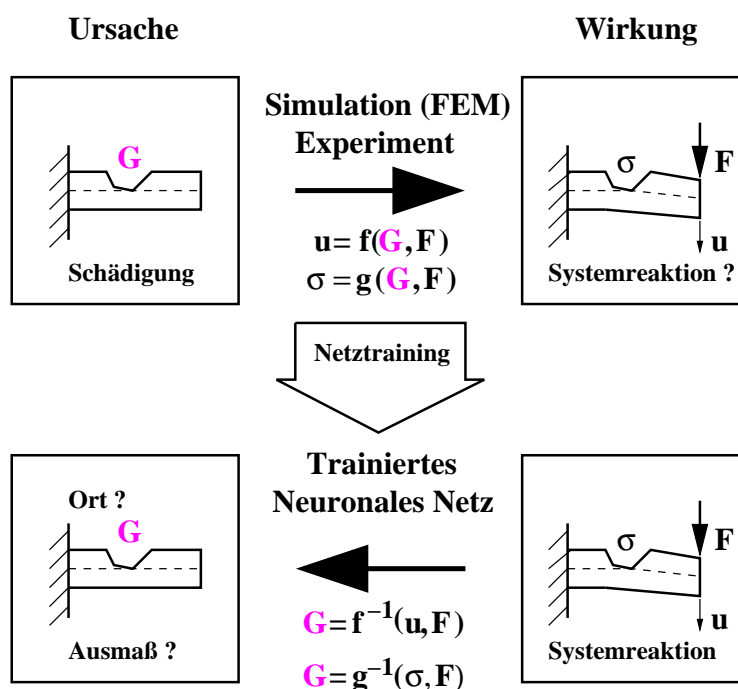


Abbildung 22: Grundprinzip Schadensanalyse

vanten Merkmale aus dem inhärenten Zusammenhang der Trainingsdaten zu extrahieren und in der Netzübertragungsfunktion zu repräsentieren.

Der Begriff Schadensklasse beschreibt die untersuchten und in den Trainingsdaten berücksichtigten Schadensarten wie z.B. Reduktion der Stabsteifigkeit (-elastizität) als Folge eines Risses. Das Analysepotential der beschriebenen Methodik ist auf die in den Trainingsdaten berücksichtigten Schadensklassen beschränkt. In dieser Arbeit werden Schädigungen der

Struktur durch eine Reduktion der Stabquerschnittsfläche über der gesamten Länge des jeweils geschädigten Stabes simuliert. Dies beschränkt das Analysepotential der hiermit trainierten Neuronalen Netze auf die Schadensklasse „reduzierte wirksame Stabquerschnittsflächen“ und damit „reduzierte Stabsteifigkeiten“. Zur Erfassung und Analyse vollkommen anderer Schädigungsarten wie z.B. Korrosion, Materialermüdung usw. müssen diese in den Test- und Trainingsdaten erfaßt sein.

Mit der in Abschnitt 6.1.2 definierten Nomenklatur folgen die Steifigkeiten bzw. Elastizitäten der kreisförmigen Stäbe zwischen den Knoten j und k zu $S_{j\ k} = E_{j\ k} A_{j\ k}$ mit Elastizitätsmodul $E_{j\ k}$ und Stabquerschnittsfläche $A_{j\ k}$. Der strukturelle Integritätskoeffizient α_i ist als Quotient der Reststeifigkeit (Restintegrität) $G_{j\ k} = (E_{j\ k} A_{j\ k})_R$ des geschädigten Elements i zwischen den Knoten j und k und $S_{j\ k} = (E_{j\ k} A_{j\ k})_N$ als Nennsteifigkeit des ungeschädigten Elements i zwischen den Knoten j und k durch

$$\alpha_i = \frac{G_{j\ k}}{S_{j\ k}} = \frac{(E_{j\ k} A_{j\ k})_R}{(E_{j\ k} A_{j\ k})_N} \quad (134)$$

definiert. Der strukturelle Integritätskoeffizient beschreibt mit reellen Werten zwischen $\alpha_i = 0$ (Totalversagen) und $\alpha_i = 1$ (keine Schädigung) den jeweiligen Schädigungszustand des Elements i relativ zum äquivalenten, ungeschädigten Bauteil i für die Schadensklasse „reduzierte Stabquerschnittsflächen“.

8.2.2 Limitierungen

Grundsätzlich muß für die erfolgreiche Analyse einer Schädigungssituation durch Neuronale Netze die Anzahl der Eingabegrößen in den Trainingsmustern mindestens gleich oder größer der Anzahl der Sollausgabegrößen sein. Ein eindeutiger Zusammenhang zwischen der Restintegrität $G_{j\ k}$ des geschädigten Stabes $Stab_{j\ k}$ einer statisch belasteten Struktur und den Knotenverschiebungen \mathbf{u} sowie den Normalspannungen $\boldsymbol{\sigma}$ der Stäbe als Systemreaktionen (siehe Abbildung 22) mit

$$G_{j\ k} = f^{-1}(\mathbf{u}, \mathbf{F}) \quad \text{bzw.} \quad G_{j\ k} = g^{-1}(\boldsymbol{\sigma}, \mathbf{F}) \quad (135)$$

ist nicht immer gegeben. Unterschiedliche Schädigungszustände des untersuchten Struktursystems können zu identischen Verschiebungsmerkmalen der Knoten bzw. Normalspannungsmustern für die Stäbe des Fachwerks führen. Zur eindeutigen Beschreibung des Strukturverhaltens der analysierten Struktur muß deshalb eine hinreichende Anzahl ausreichend unterschiedlicher Merkmalsarten als Eingabemuster verfügbar sein. Sollten Mehrdeutigkeiten auftreten, so können zusätzliche Informationen zum Strukturverhalten die verfügbaren Eingabedaten des Neuronalen Netzes ergänzen. Dies wird ausgehend von Verschiebungsmerkmalen

durch Hinzunahme von Normalspannungsmustern im Abschnitt 8.4 für statisch unbestimmte Fachwerke untersucht.

8.2.3 Test- und Trainingsdaten

Durch Vertauschung von Eingabevektoren (Ursache) und Sollausgabevektoren (Wirkung) ergeben sich die zur Approximation des gesuchten inversen Zusammenhangs notwendigen Datensysteme. Die Korrelation zwischen der zur Schadensanalyse gesuchten Ursache in Abhängigkeit von der hierdurch hervorgerufenen Wirkung bleibt auch nach der Umkehrung als inhärenter Zusammenhang in den Test- und Trainingsdaten erhalten. Es existieren verschiedene Möglichkeiten, Datensysteme (Test- und Trainingsdaten) zur Schadensanalyse von Strukturen zu erzeugen:

- Aufzeichnung im Betrieb auftretender Schädigungen und zugehöriger Wirkungen
- Experimentelle Bestimmung der Wirkungen als Reaktion auf definierte Schädigungszustände der Struktur
- Berechnung von Ursache-Wirkungs-Zusammenhängen über FE-Methoden

Die ersten beiden Möglichkeiten sind äußerst zeitintensiv und für grundlegende Untersuchungen zur Realisierbarkeit des vorgeschlagenen Algorithmus zu aufwendig. Die über Dehnmeßstreifen-Messungen ermittelbaren Dehnungen sind lokale Angaben, die nach entsprechender Transformation nur ungenau dem in den Trainingsdaten angenommenen, über der gesamten Stablänge konstanten Mittelwert der Normalspannung Rechnung tragen. Im Gegensatz dazu ist es aufwendiger, in realen technischen Systemen Knotenverschiebungen zu messen. Die Meßergebnisse sind dabei aber exakt.

Zur Generierung der Datensysteme werden Ergebnisse von Simulationsrechnungen mit dem in Abschnitt 5.3.2 beschriebenen FEM-Programm FEA, gesteuert durch den zugehörigen Permutationsalgorithmus, verwendet. Der Permutationsalgorithmus stellt sicher, daß alle Kombinationsmöglichkeiten festgelegter diskreter Schädigungszustände der Einzelstäbe für die Generierung der Datensysteme berücksichtigt sind. Dies bedeutet, daß nicht nur Einzelschädigungen, sondern auch alle Kombinationen gleichzeitig auftretender Schädigungen aller Stäbe des Fachwerks für Training, *Pruning* und Test der Neuronalen Netze zur Verfügung stehen. Auf diese Weise können Systemantworten der statisch belasteten Struktur als Reaktion auf Schädigungen an definierten Positionen der Struktur (Ort) und mit definierter Restintegrität (Ausmaß) zugehörig zur untersuchten Schadenklasse berechnet werden. Dies

ermöglicht systematische Parametervariationen für definierte Wertebereiche der untersuchten Schadensklasse und trägt so dem Umstand Rechnung, daß Schädigungen an einer beliebigen Stelle jedes Stabes der Struktur auftreten können.

Die statischen Lasten der Struktur sind für alle durch den Permutationsalgorithmus untersuchten Strukturzustände konstant und brauchen damit nicht zur Beschreibung des aktuellen Strukturzustands in den Datensystemen berücksichtigt werden, da dies keine neuen Informationen liefert. Der Elastizitätsmodul aller Stäbe des untersuchten Fachwerks wird ebenfalls als konstant angenommen. Die Umkehrung der Test- und Trainingsdaten hin zum Wirkungs-Ursache-Zusammenhang bedingt folgenden Aufbau der Eingabe- und zugehörigen Sollausgabemuster:

- Eingabemuster: Die Verschiebungen sämtlicher Knotenfreiheitsgrade sowie nach Bedarf zusätzlich normierte Normalspannungen der Stäbe
- Sollausgabemuster: Die strukturellen Integritätskoeffizienten α_i für alle $i = 1, \dots, s$ Stäbe

Als Resultat aus der Zusammensetzung der Datensysteme entspricht die Vorgehensweise entweder ausschließlich der nichtinvasiven Methode oder einer Kombination aus nichtinvasiver und direkter Methode.

Für die Erzeugung der Test- und Trainingsdaten gelten die Materialkenndaten der Al-Legierung 3.4354. Die Durchmesser der ungeschädigten Stäbe (Nenndurchmesser) sind für alle Stäbe individuell festgelegt. Alle Stäbe sind, ausgehend von individuellen Nenndurchmessern und in der durch Permutation festgelegten Reihenfolge, sukzessiv mit identischen Schädigungsniveaus belegt. Diese reduzieren die Stabquerschnittsflächen relativ zu den individuellen Nenndurchmessern.

Die Verschiebung eines Knotens i in x -Richtung ist durch $\delta_{Kn\ i,x}$ gekennzeichnet. Die Variable $\sigma_{j\ k}$ bezeichnet die aktuelle Normalspannung im Stab $j\ k$. Die Normierung der Rohtrainingsdaten erfolgt über die in Gleichung (83) beschriebene lineare Transformation. Die verwendeten Normierungszielbereiche $]0.05, 0.95[$ und $] - 0.95, 0.95[$ müssen für zusammengehörige Test- und Trainingsdaten identisch sein und sind auf die eingesetzten Transferfunktionen der Neuronen (logistische Aktivierungsfunktion und Aktivierungsfunktion Tangens hyperbolicus) abgestimmt. Deren Zusammenwirken sowie ihr Einfluß auf den Lernfortschritt und damit auf das Analysepotential des trainierten Netzwerks wird in den Abschnitten 8.3 und 8.4 untersucht.

8.2.4 Netztopologien

Die in Abschnitt 5.2.3 besprochenen problemspezifischen Neuronale Netze sind für Untersuchungen zur Schadensanalyse in Fachwerkstrukturen nicht geeignet. Dies liegt sowohl in der je nach Schädigungsgrad nicht mehr gegebenen Winkeltreue der geschädigten Strukturen als auch an nicht verfügbaren Vorabinformationen zur Problemstellung bzw. zum gesuchten Zusammenhang. Für die trotzdem untersuchten problemspezifischen Netztopologien z.B. mit Polynom- und Potenzproduktansätzen konnten keine verwertbaren Trainings- oder *Pruning*-Ergebnisse erzielt werden.

Die Approximation des zur Schadensanalyse notwendigen inversen Zusammenhangs erfolgt deshalb über konventionelle, ebenebene vollständig vernetzte *Feedforward*-Netzwerke. Nachteil dieser Vorgehensweise ist die Unübersichtlichkeit der Netzübertragungsfunktion (siehe Abschnitt 5.2.2). Die Transferfunktionen aller Neuronen der in den Abschnitten 8.3 und 8.4 eingesetzten Ausgangsnetztopologien sind durch Gleichung (78) für $k = 1, 2$ gegeben. Zur Anpassung der Ausgangsnetztopologien an die jeweilige Problemstellung werden zwei unterschiedliche Grundkonzepte untersucht:

- Training beschränkt auf die Anpassung der Verbindungsgewichte
- *Pruning* beschränkt auf verdeckte Neuronen und überflüssige Verbindungen des gesamten Netzplans mit anschließendem Nachtraining der verbliebenen Verbindungsgewichte

Neben reinem Training durch die Lernalgorithmen BackpropMomentum und Rprop kommt deshalb zusätzlich die Ausdünnung der Ausgangsnetztopologien durch das *Pruning*-Verfahren OBS (siehe Abschnitt 5.5.2) jeweils mit den untergeordneten Lernverfahren BackpropMomentum bzw. Rprop zum Einsatz.

8.3 Statisch bestimmte Fachwerke

Das in Abbildung 23 dargestellte Fachwerk FW3 dient der Validierung des vorgestellten Ansatzes zur Schadensanalyse von Fachwerkstrukturen durch konventionelle Neuronale Netze für die Problemklasse statisch bestimmter Fachwerke.

Der Aufbau der Rohdaten und ihre Normierungsintervalle zur Generierung der vier in diesem Abschnitt untersuchten Datensysteme T_{12} , T_{13} , T_{14} und T_{15} für Training, *Pruning* und Test der Neuronale Netze sind in den Tabellen 17 und 18 erfaßt. Die festgelegten Integritätskoeffizienten α_i ergeben sich z.B. für den Wert 0.64 bzw. 0.81 aus der Reduktion aller

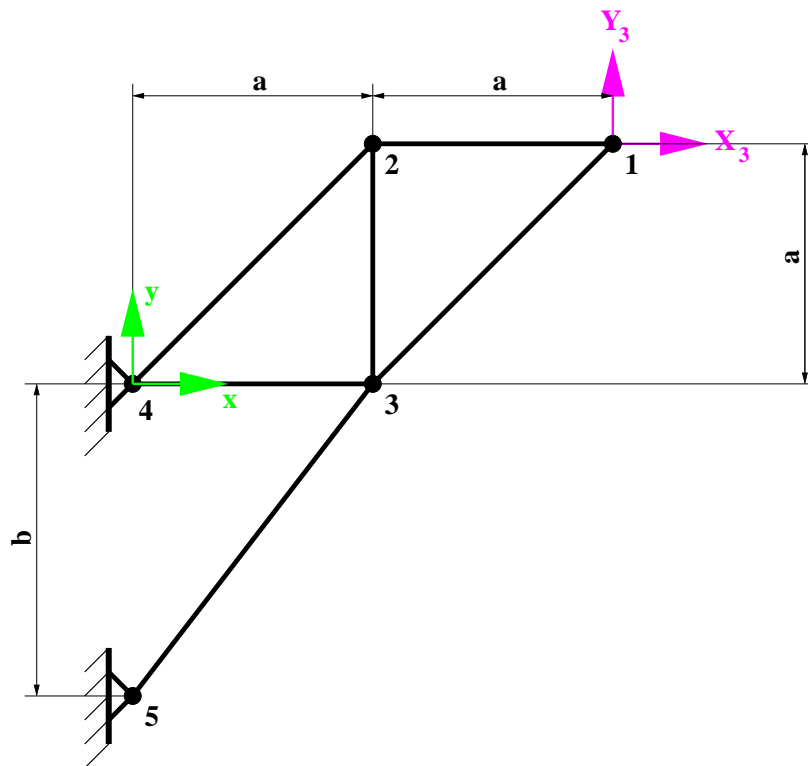


Abbildung 23: FW3: Statisch bestimmtes Fachwerk mit sechs Stäben

Stabdurchmesser des untersuchten Fachwerks auf 80 % bzw. 90 % der ungeschädigten individuellen Nenndurchmesser. Dies simuliert die für Training, *Pruning* und Test der Ausgangsnetztopologie angenommenen Schädigungen der in dieser Arbeit untersuchten Schadensklasse „reduzierte Stabquerschnittsflächen“. Die Normierungsspannung σ_N ist zu $\sigma_N = 100 \frac{\text{N}}{\text{mm}^2}$ festgelegt. Die Datensysteme T_{12} , T_{13} und T_{14} in Tabelle 17 arbeiten mit der nichtinvasiven Methode (Systemreaktion: Knotenverschiebungen) zur Bestimmung des gesuchten inversen Zusammenhangs, während das Datensystem T_{15} in Tabelle 18 die direkte Methode nutzt (Systemreaktion: Stabspannungen). Siehe hierzu Abschnitt 8.1.

Für die Analyse der Schädigungszustände des untersuchten Fachwerks FW3 sind sechs Integritätskoeffizienten und als Folge mindestens sechs unabhängige Eingabeinformationen notwendig. Die hierzu generierten konventionellen $(6 - 12 - 12 - 6)$ -Ausgangsnetztopologien ANT I und ANT J sind bzgl. ihres Netzplans identisch aufgebaut und beispielhaft in Abbildung 24 dargestellt. Die Neuronen der Ausgangsnetztopologie ANT I gehorchen der Transferfunktion aus Gleichung (78) mit $f_{\text{akt},1}(x) = \frac{1}{1+e^{-x}}$, die Neuronen der Ausgangsnetztopologie ANT J gehorchen der Transferfunktion aus Gleichung (78) mit $f_{\text{akt},2}(x) = \tanh(x)$.

Weitere untersuchte konventionelle Neuronale Netze sind die $(6 - 12 - 6)$ -Netztopologie, die

Test- und Trainingsdatensysteme T_{12}, T_{13}, T_{14}		
Eingabevektor	in	$\delta_{Kn\ 1,x}, \delta_{Kn\ 1,y}, \delta_{Kn\ 2,x}, \delta_{Kn\ 2,y}, \delta_{Kn\ 3,x}, \delta_{Kn\ 3,y}$
Sollausgabevektor	o	$\alpha_1 = \frac{(E_{1\ 2}\ A_{1\ 2})_R}{(E_{1\ 2}\ A_{1\ 2})_N}, \alpha_2 = \frac{(E_{2\ 3}\ A_{2\ 3})_R}{(E_{2\ 3}\ A_{2\ 3})_N}, \alpha_3 = \frac{(E_{1\ 3}\ A_{1\ 3})_R}{(E_{1\ 3}\ A_{1\ 3})_N},$ $\alpha_4 = \frac{(E_{2\ 4}\ A_{2\ 4})_R}{(E_{2\ 4}\ A_{2\ 4})_N}, \alpha_5 = \frac{(E_{3\ 4}\ A_{3\ 4})_R}{(E_{3\ 4}\ A_{3\ 4})_N}, \alpha_6 = \frac{(E_{3\ 5}\ A_{3\ 5})_R}{(E_{3\ 5}\ A_{3\ 5})_N}$
Trainingsdaten	T_{12} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 729 Mustern 0.64, 0.81, 1.0]0.05, 0.95[
Testdaten	T_{12t} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 64 Mustern 0.7056, 0.8464]0.05, 0.95[
Trainingsdaten	T_{13} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 729 Mustern 0.64, 0.81, 1.0] - 0.95, 0.95[
Testdaten	T_{13t} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 64 Mustern 0.7056, 0.8464] - 0.95, 0.95[
Trainingsdaten	T_{14} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 46656 Mustern 0.5625, 0.64, 0.7225, 0.81, 0.9025, 1.0] - 0.95, 0.95[
Testdaten	T_{14t} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 64 Mustern 0.680625, 0.855625] - 0.95, 0.95[

Tabelle 17: Datensysteme T_{12}, T_{13} und T_{14} zu ANT I und ANT J

dreidimensionale $(6 \times 5 - 6 \times 6 - 6 \times 6 - 6)$ -Netztopologie sowie die $(6 \times 5 - 6 \times 5 - 6)$ -Netztopologie mit einer Matrix von 6×5 Neuronen der Eingabeschicht, 6×5 verdeckten Neuronen und 6 Neuronen der Ausgabeschicht. Die durchgeführten Simulationsrechnungen mit verschiedenen Parametervariationen zeigen deutlich schlechteres Lern- und Generalisierungsverhalten im Vergleich zu den dargestellten Ergebnissen der Ausgangsnetztopologien ANT I und ANT J. Für die folgenden Untersuchungen werden deshalb die Ausgangsnetztopologien der Typen ANT I bzw. ANT J in identischer bzw. erweiterter Form eingesetzt.

Test- und Trainingsdatensystem T ₁₅	
Eingabevektor	in $\frac{\sigma_{1,2}}{\sigma_N}, \frac{\sigma_{2,3}}{\sigma_N}, \frac{\sigma_{1,3}}{\sigma_N}, \frac{\sigma_{2,4}}{\sigma_N}, \frac{\sigma_{3,4}}{\sigma_N}, \frac{\sigma_{3,5}}{\sigma_N}$
Sollausgabevektor	o $\alpha_1 = \frac{(E_{1,2} A_{1,2})_R}{(E_{1,2} A_{1,2})_N}, \alpha_2 = \frac{(E_{2,3} A_{2,3})_R}{(E_{2,3} A_{2,3})_N}, \alpha_3 = \frac{(E_{1,3} A_{1,3})_R}{(E_{1,3} A_{1,3})_N},$ $\alpha_4 = \frac{(E_{2,4} A_{2,4})_R}{(E_{2,4} A_{2,4})_N}, \alpha_5 = \frac{(E_{3,4} A_{3,4})_R}{(E_{3,4} A_{3,4})_N}, \alpha_6 = \frac{(E_{3,5} A_{3,5})_R}{(E_{3,5} A_{3,5})_N}$
Trainingsdaten	T ₁₅ $\Psi = 1$ mit 729 Mustern Wertebereich α_i 0.64, 0.81, 1.0 Normierungsintervall]0.05, 0.95[
Testdaten	T _{15t} $\Psi = 1$ mit 64 Mustern Wertebereich α_i 0.7056, 0.8464 Normierungsintervall]0.05, 0.95[

Tabelle 18: Datensystem T₁₅ zu ANT I und ANT J

8.3.1 Training der Ausgangsnetztopologien

Die durchgeführten Simulationsrechnungen zu Training und Test der Ausgangsnetztopologien ANT I und ANT J für die Analyse von Schädigungen der Fachwerkstruktur FW3 sind in Tabelle 19 zusammengefaßt. Der beim *Recall* der Testdaten erzielte Gesamttestfehler μ_{test} ergibt sich als Mittelwert über alle Stäbe und alle Muster der Testdaten und ist in Gleichung (86) definiert. Tabelle 20 zeigt repräsentative Testresultate erfolgreicher Trainingsrechnungen aus

Parameter Training ANT I und ANT J			
ANT	I, J	Trainingsalgorithmen	BackpropMomentum
Trainingsdaten	T ₁₂ , T ₁₃ , T ₁₄ , T ₁₅		Rprop
Testdaten	T _{12t} , T _{13t} , T _{14t} , T _{15t}	Initialisierungsintervalle	[0.0, 1.0], [-1.0, 1.0]

Tabelle 19: Parameter Training ANT I und ANT J

den durchgeführten, umfangreichen Variationsrechnungen hinsichtlich Lernparameter, Neuronenfunktionalität, Lernalgorithmus sowie Zusammensetzung der Trainingsdaten. Die Testergebnisse beziehen sich auf den Test (*Recall*) der trainierten Netztopologie mit den angegebenen Testdaten. Die Abkürzung (kbE) in den Ergebnistabellen kennzeichnet Fälle, bei denen aus den untersuchten Parametervariationen keine brauchbaren Ergebnisse erzielt werden konnten. Die in Tabelle 20 zusammengefaßten Ergebnisse verdeutlichen die Leistungsfähigkeit erfolgreich trainierter, konventioneller Neuronaler Netze für die Analyse unbekannter

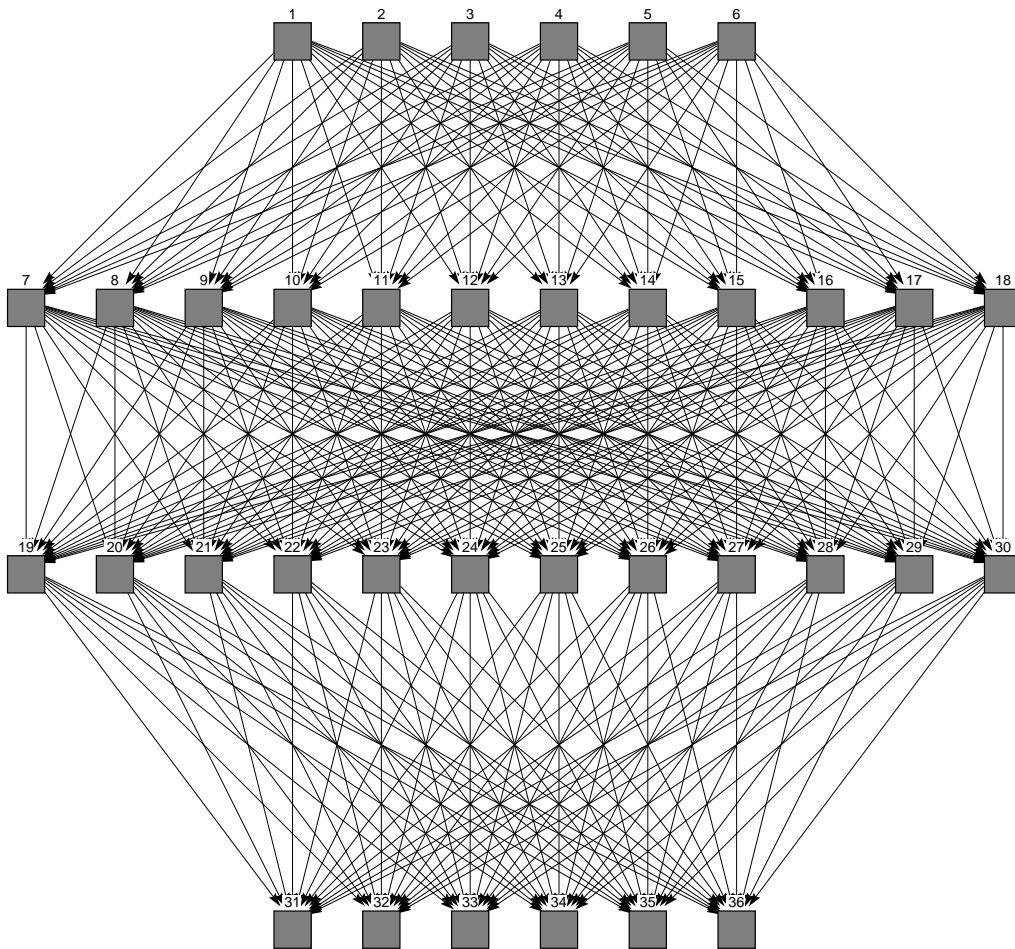


Abbildung 24: ANT I und ANT J

Schädigungen der untersuchten, statisch bestimmten Fachwerkstruktur. Dies gilt auch für das Datensystem T_{14} mit dem großen Wertebereich trainierter Schädigungszustände.

Die Fälle mit nicht verwertbaren oder sehr schlechten Ergebnissen beschränken sich für beide Lernverfahren auf die Ausgangsnetztopologie ANT I mit der zur Transferfunktion in Gleichung (78) gehörenden Aktivierungsfunktion $f_{\text{akt},1}(x) = \frac{1}{1+e^{-x}}$ und die Datensysteme T_{13} und T_{14} . Damit sind die Ergebnisse dem für die Aktivierungsfunktion $f_{\text{akt},1}(x)$ nicht optimalen Normierungsintervall $]-0.95, 0.95[$ zuzuordnen. Der im folgenden verwendete Begriff „reines Trainingsergebnis“ beschreibt Resultate Neuronaler Netze, die nur durch Training und nicht durch eine Kombination von *Pruning* mit untergeordnetem Training erzielt werden.

Ergebnisse Training ANT I und ANT J				
ANT	Daten		Lernalgorithmus	Test in [%]
	Test	Training		μ_{test}
ANT I	T _{12t}	T ₁₂	BackpropMomentum	9.03
ANT J	T _{12t}	T ₁₂	BackpropMomentum	11.64
ANT I	T _{12t}	T ₁₂	Rprop	11.01
ANT J	T _{12t}	T ₁₂	Rprop	8.95
ANT I	T _{13t}	T ₁₃	BackpropMomentum	kbE
ANT J	T _{13t}	T ₁₃	BackpropMomentum	3.05
ANT I	T _{13t}	T ₁₃	Rprop	kbE
ANT J	T _{13t}	T ₁₃	Rprop	3.36
ANT I	T _{14t}	T ₁₄	BackpropMomentum	kbE
ANT J	T _{14t}	T ₁₄	BackpropMomentum	3.21
ANT I	T _{14t}	T ₁₄	Rprop	kbE
ANT J	T _{14t}	T ₁₄	Rprop	6.04

Tabelle 20: Ergebnisse Training ANT I und ANT J

8.3.2 Reduktion der Ausgangsnetztopologien

Einen Überblick über die durchgeführten Simulationsrechnungen (*Pruning*) zur Untersuchung des Analysepotentials der Ausgangsnetztopologien ANT I und ANT J gibt Tabelle 21. Tabelle 22 faßt die Simulationsergebnisse zusammen. Bei den dargestellten Testergebnissen

Pruning ANT I und ANT J	
(Recall) ANT	I, J
Trainingsdaten	T ₁₂ , T ₁₃ , T ₁₅
Testdaten	T _{12t} , T _{13t} , T _{15t}
<i>Pruning</i> -Algorithmen	OBS mit Backprop Momentum OBS mit Rprop
Initialisierungsintervalle	[0.0, 1.0], [-1.0, 1.0]

Tabelle 21: Parameter *Pruning* ANT I und ANT J

handelt es sich um erfolgreiche *Pruning*-Resultate. Die Angaben zur Reduktion der Netztopologie beschreiben die Anzahl verdeckter Neuronen (*units*) und Verbindungen (*links*) des gesamten Netzplans, die der *Pruning*-Algorithmus gegenüber der Ausgangsnetztopologie löscht. Die umfangreichen, zugrundeliegenden Simulationsrechnungen zur Parametervariation betreffen die Parameter der Lernalgorithmen, die Neuronenfunktionalität, den OBS untergeordneten Lernalgorithmus sowie das verwendete Datensystem. Die dabei erzielten Ergebnisse sind in

Ergebnisse <i>Pruning</i>						
ANT	Daten		<i>Pruning</i> -Algorithmus	Test in [%]	Reduktion in [%]	
	Test	Training		μ_{test}	<i>links</i>	<i>units</i>
ANT I	T _{12t}	T ₁₂	OBS mit BackpropMomentum	7.13	81.94	12.5
ANT J	T _{12t}	T ₁₂	OBS mit BackpropMomentum	8.83	80.56	25
ANT I	T _{12t}	T ₁₂	OBS mit Rprop	9.87	57.99	4.17
ANT J	T _{12t}	T ₁₂	OBS mit Rprop	11.00	2.78	0
ANT I	T _{13t}	T ₁₃	OBS mit BackpropMomentum	kbE	—	—
ANT J	T _{13t}	T ₁₃	OBS mit BackpropMomentum	2.96	82.99	16.67
ANT I	T _{13t}	T ₁₃	OBS mit Rprop	kbE	—	—
ANT J	T _{13t}	T ₁₃	OBS mit Rprop	3.01	84.72	33.33

Tabelle 22: Ergebnisse *Pruning* ANT I und ANT J

Tabelle 22 dargestellt und zeigen deutlich das Reduktionspotential der Ausgangsnetztopologien. Die z.T. signifikant reduzierten Resttopologien zeigen vergleichbare Generalisierungsleistungen im Vergleich zu reinen Trainingsergebnissen. Die Fälle mit nicht verwertbaren (kbE) oder sehr schlechten Ergebnissen beschränken sich für die beiden untergeordneten Lernverfahren auf die Ausgangsnetztopologie ANT I mit der zur Transferfunktion in Gleichung (78) gehörenden Aktivierungsfunktion $f_{\text{akt},1}(x) = \frac{1}{1+e^{-x}}$ sowie das Datensystem T₁₃. Damit sind die Ergebnisse dem für die Aktivierungsfunktion $f_{\text{akt},1}(x)$ nicht optimalen Normierungsintervall $]-0.95, 0.95[$ zuzuordnen.

Am Beispiel des Datensystems T₁₅ sollen Trainings- und *Pruning*-Ergebnisse für gleichartige Parametereinstellungen hinsichtlich des dabei erzielten Generalisierungsergebnisses direkt verglichen werden. Die in Tabelle 23 jeweils in einer Zeile dargestellten Ergebnisse zu Training und *Pruning* gehören zu gleichartigen Parametereinstellungen hinsichtlich des reinen Lernalgorithmus', des OBS untergeordneten Lernalgorithmus' sowie zu identischen Ausgangsnetztopologien. Der Vergleich zeigt, daß auch mit signifikant reduzierten Netztopologi-

Vergleich Training und Pruning zu Datensystem T ₁₅						
ANT	Pruning-Algorithmus	Reduktion in [%]		Test in [%]		Trainingsalgorithmus
		<i>links</i>	<i>units</i>	μ_{test}	μ_{test}	
ANT I	OBS mit BackpropMomentum	75.35	12.5	3.56	2.03	BackpropMomentum
ANT J	OBS mit BackpropMomentum	80.56	20.83	4.52	2.48	BackpropMomentum
ANT I	OBS mit Rprop	90.28	33.33	1.95	2.23	Rprop
ANT J	OBS mit Rprop	93.75	50	2.05	2.15	Rprop

Tabelle 23: Vergleich Training und *Pruning* für Datensystem T₁₅ zu ANT I und ANT J

en vergleichbare Testergebnisse erzielbar sind. Das *Pruning*-Verfahren OBS mit untergeordnetem Lernverfahren Rprop zeigt dabei bessere Ergebnisse als OBS mit BackpropMomentum. Abbildung 25 zeigt die durch OBS mit Rprop reduzierte Ausgangsnetztopologie ANT J zugehörig zum Datensystem T₁₅ (siehe Tabelle 23). Gegenüber der Ausgangsnetztopologie sind 93.75 % der *links* und 50 % der verdeckten Neuronen durch das *Pruning*-Verfahren gelöscht worden. Der mittlere Testfehler der so reduzierten Netztopologie liegt bei $\mu_{\text{test}} = 2.05$ % und damit um 0.1 % unter dem Testfehler des reinen Trainingsergebnisses mit Rprop.

Ein Vergleich der Ergebnisse zwischen Tabelle 20 zu reinen Trainingsergebnissen und Tabelle 22 zu *Pruning*-Ergebnissen zeigt bei vergleichbarem Fehlerniveau ein signifikantes Reduktionspotential bei *links* und verdeckten Neuronen gegenüber den Ausgangsnetztopologien ANT I und ANT J. Der Mittelwert des Gesamttestfehlers μ_{test} und der Netzfehler SSE (siehe Gleichung (85)) nach erfolgreichem Training bzw. *Pruning* der Ausgangsnetztopologie zeigen proportionales Verhalten: μ_{test} sinkt monoton mit abnehmendem SSE. Als repräsentative Beispiele seien zwei Ergebnisse aus dem Training der Ausgangsnetztopologie ANT I mit dem Datensystem T₁₅ und dem Lernalgorithmus BackpropMomentum genannt:

- $\mu_{\text{test}} = 2.03$ % zugehörig zu SSE = 0.00331 nach 100 000 Epochen
- $\mu_{\text{test}} = 9.66$ % zugehörig zu SSE = 0.63217 nach 100 000 Epochen (nicht in Tabelle 23)

Dies gilt in gleicher Weise für die untersuchten Datensysteme T₁₂, T₁₃, T₁₄ und T₁₅, insbesondere für Trainings- und in abgeschwächter Form auch für *Pruning*-Resultate.

Der folgende Abschnitt untersucht die Anwendbarkeit der bisher gewonnenen Erkenntnisse auf statisch unbestimmte Fachwerke. Besondere Beachtung findet dabei eine ausreichende Verteilung der Trainingsdaten.

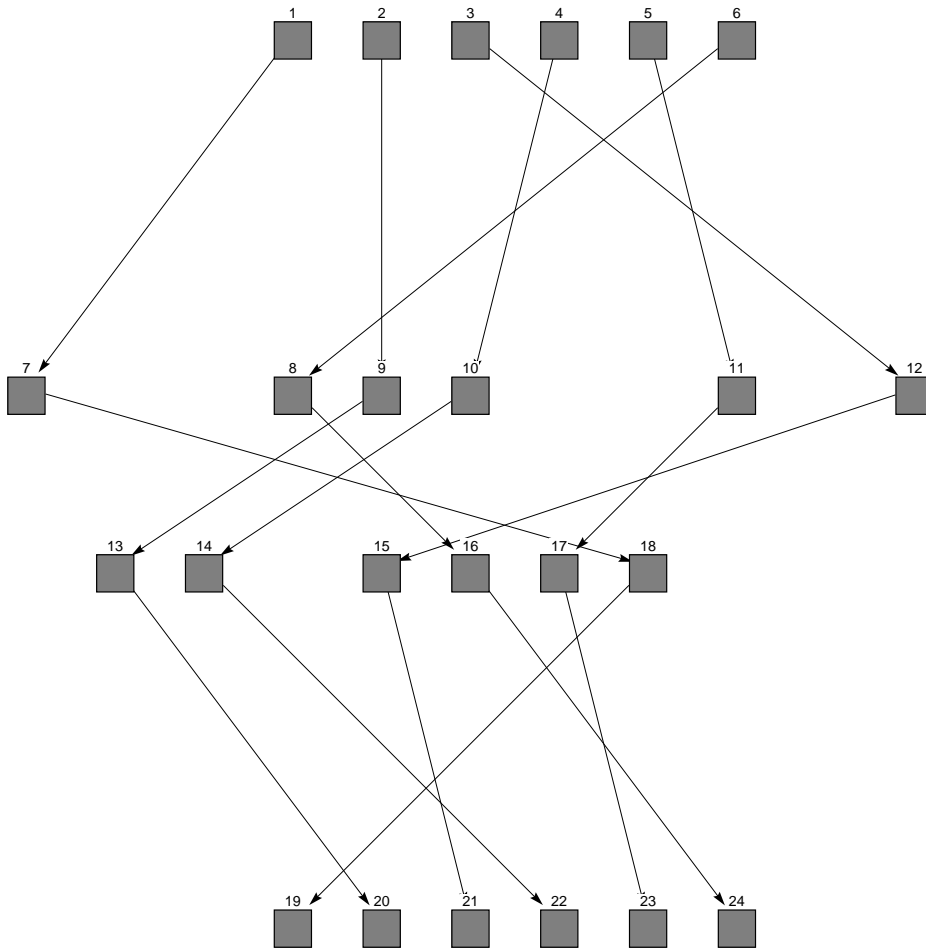


Abbildung 25: Reduzierte Netztopologie NT sep8d

8.4 Statisch unbestimmte Fachwerke

In diesem Abschnitt sollen die Möglichkeiten konventioneller Neuronaler Netze bei der Schadensanalyse statisch unbestimmter Fachwerke untersucht werden. Als Anwendungsbeispiel dient das in Abbildung 26 dargestellte, einfach statisch unbestimmte Fachwerk FW4. Erste Machbarkeitsstudien anhand von Einzelschädigungen des Stabes $Stab_1$ $_2$ bei intakter Reststruktur sowie anhand von Schädigungen der Stäbe $Stab_1$ $_2$ und $Stab_2$ $_4$ bei intakter Reststruktur bestätigen die Realisierbarkeit dieser Grundidee.

Da für die sechs unbekanntem Integritätskoeffizienten der Stäbe des Fachwerks FW4 nur fünf freie Knotenverschiebungen als Systemreaktionen für die Eingabevektoren der Test- und Trainingsdaten zur Verfügung stehen, sind weitere Informationen zur Erfassung des aktuellen Systemzustands als Reaktion auf die Belastungs- und Schädigungssituation notwendig. Für die Eingabevektoren werden deshalb entweder die Normalspannungen aller Stäbe einzeln

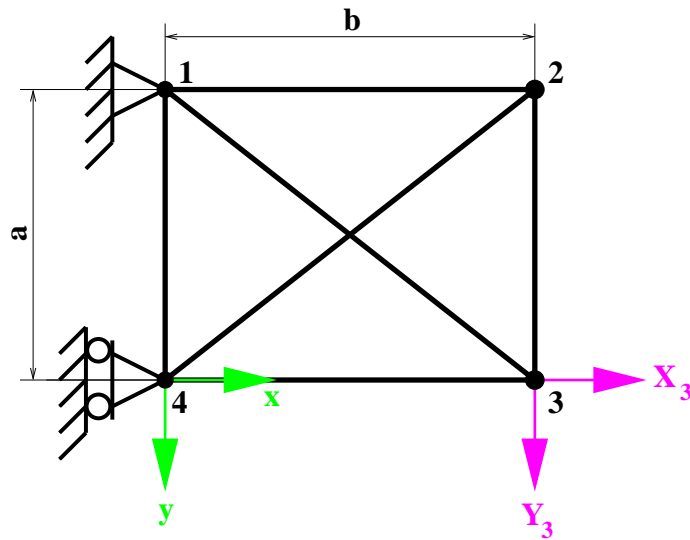


Abbildung 26: FW4: Statisch unbestimmtes Fachwerk mit sechs Stäben

(siehe Tabelle 24) oder in Kombination mit den freien Knotenverschiebungen des Fachwerks FW4 (siehe Tabelle 25) berücksichtigt. Damit nutzen die Trainingsmuster die direkte Methode sowie eine Kombination aus direkter und nichtinvasiver Methode zur Bestimmung des gesuchten inversen Zusammenhangs.

Wie in Abschnitt 8.2.3 beschrieben, berücksichtigt der Permutationsalgorithmus des FEM-Programms FEA sämtliche Schädigungskombinationen aller Stäbe für die untersuchten Daten T_{17} (siehe Tabelle 24) und T_{16} (siehe Tabelle 25). Die Normierungsgrößen $\sigma_N = 10 \frac{N}{\text{mm}^2}$ als

Test- und Trainingsdatensystem T_{17}		
Eingabevektor	in	$\frac{\sigma_{1,2}}{\sigma_N}, \frac{\sigma_{2,3}}{\sigma_N}, \frac{\sigma_{3,4}}{\sigma_N}, \frac{\sigma_{1,4}}{\sigma_N}, \frac{\sigma_{1,3}}{\sigma_N}, \frac{\sigma_{2,4}}{y\sigma_N}$
Sollausgabevektor	o	$\alpha_1 = \frac{(E_{1,2} A_{1,2})_R}{(E_{1,2} A_{1,2})_N}, \alpha_2 = \frac{(E_{2,3} A_{2,3})_R}{(E_{2,3} A_{2,3})_N}, \alpha_3 = \frac{(E_{3,4} A_{3,4})_R}{(E_{3,4} A_{3,4})_N},$ $\alpha_4 = \frac{(E_{1,4} A_{1,4})_R}{(E_{1,4} A_{1,4})_N}, \alpha_5 = \frac{(E_{1,3} A_{1,3})_R}{(E_{1,3} A_{1,3})_N}, \alpha_6 = \frac{(E_{2,4} A_{2,4})_R}{(E_{2,4} A_{2,4})_N}$
Trainingsdaten	T_{17} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 15625 Mustern 0.9216, 0.9409, 0.9604, 0.9801, 1.0]0.05, 0.95[
Testdaten	T_{17t} Wertebereich α_i Normierungsintervall	$\Psi = 1$ mit 64 Mustern 0.950625, 0.970225]0.05, 0.95[

Tabelle 24: Datensystem T_{17} zu ANT I und ANT J

Normierungsspannung bzw. $\delta_N = 1\text{mm}$ als Normierungslänge bewirken numerisch ausgeglichene bzw. dimensionslose Daten.

Test- und Trainingsdatensystem T_{16}		
Eingabevektor	in	$\frac{\delta_{Kn\ 2,x}}{\delta_N}, \frac{\delta_{Kn\ 2,y}}{\delta_N}, \frac{\delta_{Kn\ 3,x}}{\delta_N}, \frac{\delta_{Kn\ 3,y}}{\delta_N}, \frac{\delta_{Kn\ 4,y}}{\delta_N},$ $\frac{\sigma_{1\ 2}}{\sigma_N}, \frac{\sigma_{2\ 3}}{\sigma_N}, \frac{\sigma_{3\ 4}}{\sigma_N}, \frac{\sigma_{1\ 4}}{\sigma_N}, \frac{\sigma_{1\ 3}}{\sigma_N}, \frac{\sigma_{2\ 4}}{\sigma_N}$
Sollausgabevektor	o	$\alpha_1 = \frac{(E_{1\ 2}\ A_{1\ 2})_R}{(E_{1\ 2}\ A_{1\ 2})_N}, \alpha_2 = \frac{(E_{2\ 3}\ A_{2\ 3})_R}{(E_{2\ 3}\ A_{2\ 3})_N}, \alpha_3 = \frac{(E_{3\ 4}\ A_{3\ 4})_R}{(E_{3\ 4}\ A_{3\ 4})_N},$ $\alpha_4 = \frac{(E_{1\ 4}\ A_{1\ 4})_R}{(E_{1\ 4}\ A_{1\ 4})_N}, \alpha_5 = \frac{(E_{1\ 3}\ A_{1\ 3})_R}{(E_{1\ 3}\ A_{1\ 3})_N}, \alpha_6 = \frac{(E_{2\ 4}\ A_{2\ 4})_R}{(E_{2\ 4}\ A_{2\ 4})_N}$
Trainingsdaten	T_{16}	$\Psi = 1$ mit 15625 Mustern Wertebereich α_i 0.9216, 0.9409, 0.9604, 0.9801, 1.0 Normierungsintervall]0.05, 0.95[
Testdaten	T_{16t}	$\Psi = 1$ mit 64 Mustern Wertebereich α_i 0.950625, 0.970225 Normierungsintervall]0.05, 0.95[

Tabelle 25: Datensystem T_{16} zu ANT K und ANT L

Die zur vorliegenden Problemstellung generierten, ebenenweise vollständig vernetzten, konventionellen (11 – 12 – 12 – 6)-Ausgangsnetztopologien ANT K und ANT L sind bzgl. ihres Netzplans gleichartig aufgebaut und beispielhaft in Abbildung 27 dargestellt. Beide Ausgangsnetztopologien unterscheiden sich nur nach ihrer Neuronenfunktionalität: Die Neuronen der Ausgangsnetztopologie ANT K gehorchen der Transferfunktion aus Gleichung (76) mit $f_{akt,1}(x) = \frac{1}{1+e^{-x}}$, die Neuronen der Ausgangsnetztopologie ANT L gehorchen der Transferfunktion aus Gleichung (77) mit $f_{akt,2}(x) = \tanh(x)$. Zusätzlich werden die aus Abschnitt 8.3 bekannten Ausgangsnetztopologien ANT I und ANT J eingesetzt.

8.4.1 Training und Reduktion der Ausgangsnetztopologien

Die Simulationsrechnungen zu den untersuchten Parametervariationen für Training und *Pruning* der Ausgangsnetztopologien ANT I, ANT J, ANT K und ANT L sind in den Tabellen 26 und 27 zusammengefasst. Die Tabellen 28 bzw. 29 vergleichen jeweils die besten sich hieraus ergebenden Generalisierungsergebnisse zugehörig zu den Datensystemen T_{17} bzw. T_{16} . Die jeweils in einer Zeile dargestellten Resultate zu Training und *Pruning* identischer Ausgangsnetztopologien mit gemeinsamen Trainingsdaten können das Resultat unterschiedlicher Parametereinstellungen hinsichtlich reinem Training und *Pruning* (OBS) mit untergeordnetem

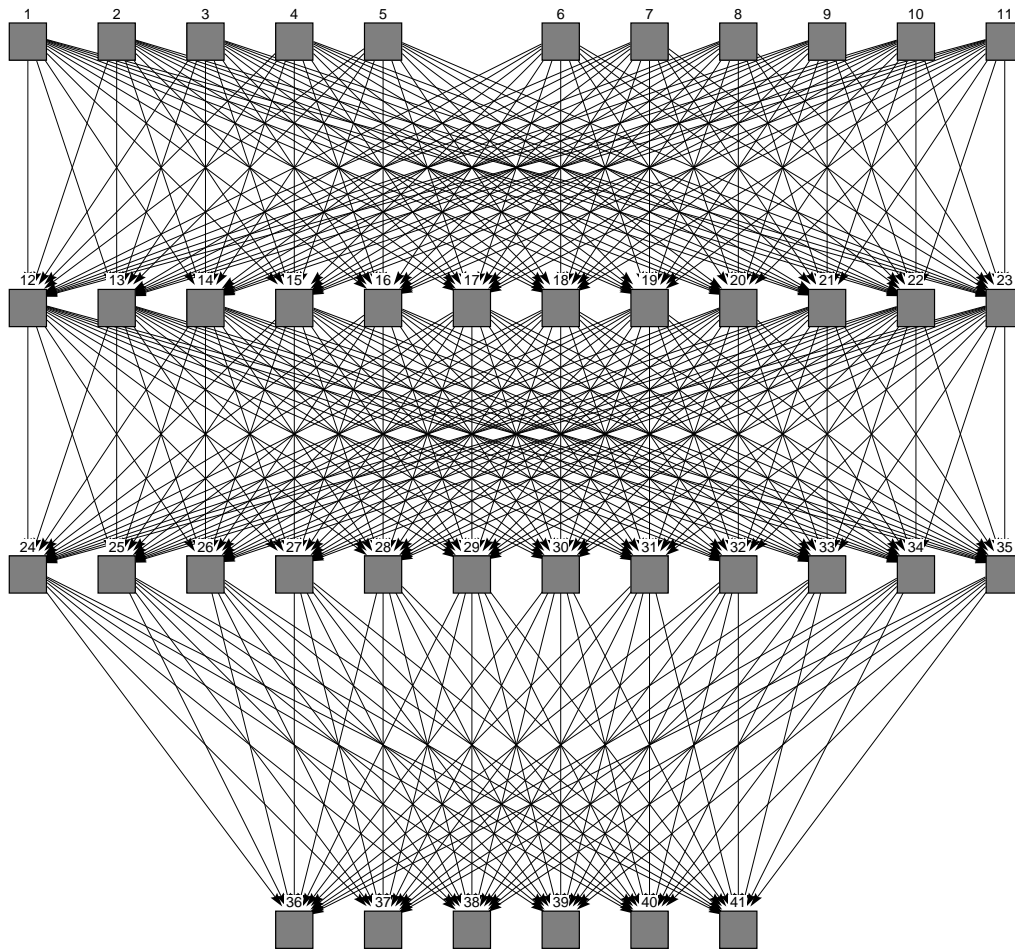


Abbildung 27: ANT K und ANT L

Training ANT I, ANT J, ANT K und ANT L			
ANT	I, J, K, L	Trainingsalgorithmen	BackpropMomentum
Trainingsdaten	T ₁₇ , T ₁₆		Rprop
Testdaten	T _{17t} , T _{16t}	Initialisierungsintervall	[-1.0,1.0]

Tabelle 26: Parameter Training ANT I, ANT J, ANT K und ANT L

Lernalgorithmus sein. Trotz der dabei erkennbaren Ergebnisschwankungen verdeutlichen die dort dargestellten Resultate, daß für das untersuchte, statisch unbestimmte Fachwerk FW4 eine Schadensanalyse durch trainierte und durch *Pruning* reduzierte Neuronale Netze möglich ist. Abbildung 28 zeigt die durch OBS mit Rprop reduzierte Ausgangsnetztopologie ANT K zugehörig zum Datensystem T₁₆ (Resttopologie NT jan39i). Gegenüber der Ausgangsnetztopologie sind dort 84.48 % der *links* und 20.83 % der verdeckten Neuronen durch das

Parameter Pruning ANT I, ANT J, ANT K und ANT L	
ANT	I, J, K, L
Trainingsdaten	T ₁₇ , T ₁₆
Testdaten	T _{17t} , T _{16t}
Pruning-Algorithmen	OBS mit Backprop Momentum OBS mit Rprop
Initialisierungsintervall	[-1.0, 1.0]

Tabelle 27: Parameter *Pruning* ANT I, ANT J, ANT K und ANT L

Vergleich Training und Pruning zu Datensystem T ₁₇						
ANT	Pruning-Algorithmus	Reduktion in [%]		Test in [%]		Trainingsalgorithmus
		<i>links</i>	<i>units</i>	μ_{test}	μ_{test}	
ANT I	OBS mit BackpropMomentum	—	—	kbE	kbE	BackpropMomentum
ANT J	OBS mit BackpropMomentum	70.14	12.5	8.39	7.14	BackpropMomentum
ANT I	OBS mit Rprop	82.64	25	8.46	6.68	Rprop
ANT J	OBS mit Rprop	1.39	0	7.85	7.12	Rprop

Tabelle 28: Vergleich Training und *Pruning* für Datensystem T₁₇ zu ANT I und ANT J

Pruning-Verfahren gelöscht. Der mittlere Testfehler der reduzierten Netztopologie NT jan39i liegt bei $\mu_{\text{test}} = 8.60$ % und damit um 1.73 % über dem Testfehler des reinen Trainingsergebnisses mit Rprop.

Vergleich Training und Pruning zu Datensystem T ₁₆						
ANT	Pruning-Algorithmus	Reduktion [%]		Test [%]		Trainingsalgorithmus
		<i>links</i>	<i>units</i>	μ_{test}	μ_{test}	
ANT K	OBS mit BackpropMomentum	51.72	0	7.96	6.88	BackpropMomentum
ANT L	OBS mit BackpropMomentum	58.91	0	8.02	11.87	BackpropMomentum
ANT K	OBS mit Rprop	84.48	20.83	8.60	6.87	Rprop
ANT L	OBS mit Rprop	2.59	0	8.23	7.58	Rprop

Tabelle 29: Vergleich Training und *Pruning* für Datensystem T₁₆ zu ANT K und ANT L

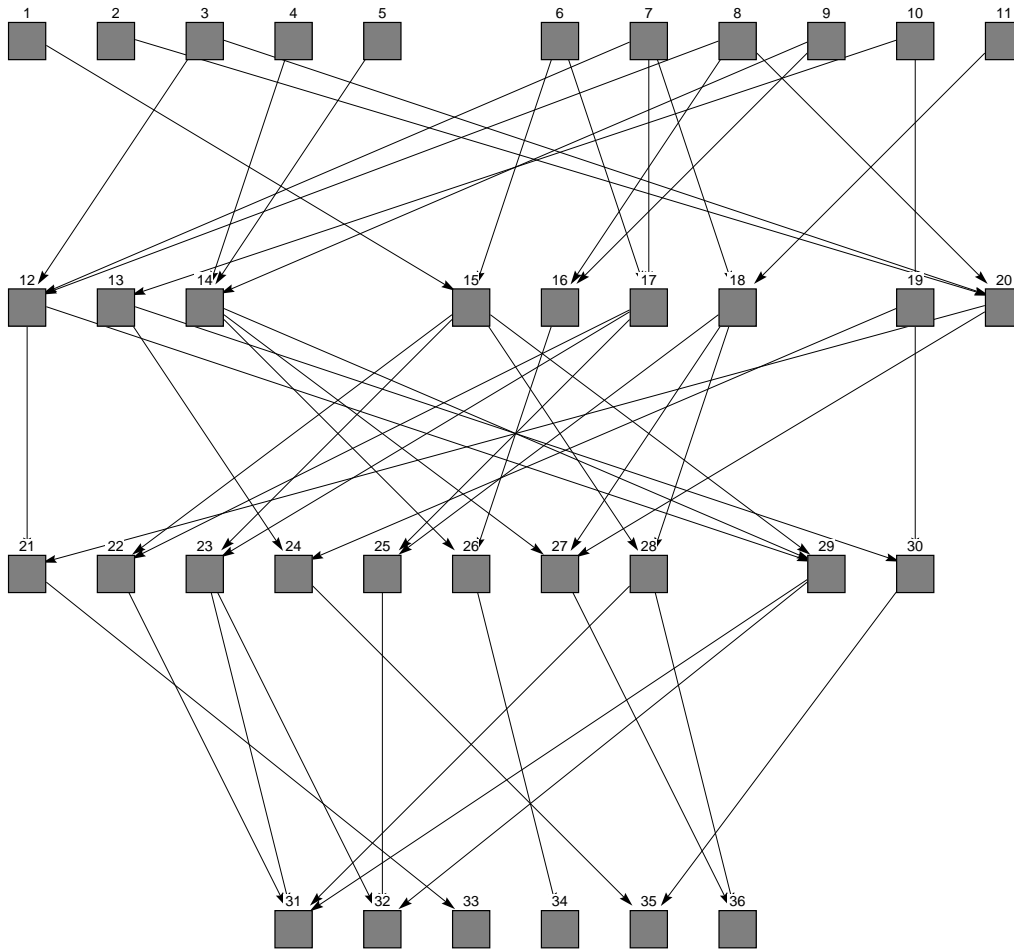


Abbildung 28: Reduzierte Netztopologie NT jan39i

Insbesondere bei nicht zu niedrigen Fehlerwerten E bzw. SSE nach Abschluß der Trainings- bzw. *Pruning*-Phase können gute Generalisierungsergebnisse hinsichtlich unbekannter Testmuster beobachtet werden. Der Mittelwert des Gesamttestfehlers μ_{test} und der Netzfehler SSE (siehe Gleichung (85)) nach Training bzw. *Pruning* der Ausgangsnetztopologien zeigen sowohl proportionales als auch umgekehrt proportionales Verhalten: Der Gesamttestfehler sinkt mit abnehmendem Wert von SSE und steigt dann mit weiter abnehmendem Wert von SSE aber wieder an. Als repräsentative Beispiele seien drei Ergebnisse aus dem Training der Ausgangsnetztopologie ANT K mit dem Datensystem T_{16} und dem Lernalgorithmus Backprop-Momentum genannt:

- $\mu_{\text{test}} = 12.34 \%$ zugehörig zu $SSE = 823.73$ nach 100 000 Epochen
- $\mu_{\text{test}} = 6.88 \%$ zugehörig zu $SSE = 762.76$ nach 100 000 Epochen

- $\mu_{\text{test}} = 35.01$ % zugehörig zu $\text{SSE} = 83.78$ nach 100 000 Epochen

Übertrainieren scheidet als Ursache aus, da alle Trainingsergebnisse mit der gleichen Anzahl an Trainingsepochen erzielt wurden. Dies gilt für die untersuchten Datensysteme T_{17} und T_{16} insbesondere für Trainings- und in abgeschwächter Form auch für *Pruning*-Ergebnisse. Grund hierfür ist die zufällige Initialisierung der Gewichtsmatrizen der Ausgangsnetztopologien bei Start der Trainings- bzw. *Pruning*-Simulationen.

8.5 Fazit

Das Grundprinzip zur Ermittlung eines inversen Zusammenhangs durch Neuronale Netze sowie die Anwendung dieses Grundprinzips auf die Schadensanalyse basieren auf Resultaten aus der Literatur und sind in Abschnitt 1.2.3 zitiert. Basierend auf den dortigen Ergebnissen erweitern die vorliegenden Untersuchungen das Anwendungsspektrum Neuronaler Netze auf den Bereich statisch unbestimmter Fachwerke. Weiterhin konnte für den Anwendungsfall Schadensanalyse erstmals das Reduktionspotential durch *Pruning*-Verfahren wie OBS gezeigt werden. Die Untersuchungen verdeutlichen das Potential intelligent reduzierter Neuronaler Netze bei der Schadensanalyse der untersuchten Fachwerkstrukturen.

Zur Schadensanalyse statisch bestimmter und unbestimmter Fachwerke werden ebenenweise vollständig vernetzte konventionelle Neuronale Netze eingesetzt. Die erfolgreich trainierten bzw. durch *Pruning* reduzierten Neuronalen Netze sind in der Lage, für das untersuchte statisch bestimmte und unbestimmte Fachwerk jeweils unbekannte Schädigungen nach Existenz, Ort und Ausmaß einer oder mehrerer gleichzeitig auftretender Schädigungen zu klassifizieren. Das Analysepotential der vorgestellten Methodik ist auf die in den Trainingsdaten berücksichtigten Schadensklassen beschränkt. Die Erweiterung der vorgestellten Systematik auf statisch unbestimmte Fachwerkstrukturen und der Einsatz von *Pruning*-Verfahren zur Ausdünnung der Ausgangsnetztopologien sind neue Beiträge der vorliegenden Arbeit.

Bei der Untersuchung des statisch unbestimmten Fachwerks FW4 konnte mit guten Generalisierungsergebnissen ein begrenzter Wertebereich erfaßter Schadensniveaus in jeweils einem Neuronalen Netz abgebildet werden. Die Abdeckung eines breiten Wertebereichs erfaßter Schadensniveaus ist durch mehrere, sich in den Wertebereichen ihrer Datensysteme ergänzender, Neuronaler Netze realisierbar. Damit ist ein breiter Parameterbereich potentieller Schädigungszustände durch mehrere gleichartige, auf schmale Einzelbereiche spezialisierte Neuronale Netze erfassbar. Die Zuordnung aktueller Systemantworten zu den einzelnen Bereichsnetzen kann über ein Expertensystem oder ein entsprechend trainiertes Neuronales Netz

erfolgen.

Für zukünftige Untersuchungen sind die zum Training Neuronaler Netze eingesetzten Systemparameter und -reaktionen besser auf die technischen Möglichkeiten einer *Online*-Schadensüberwachung anzupassen. Als Beispiel sollte untersucht werden, ob und in welchem Ausmaß die bei statisch unbestimmten Fachwerken zusätzlich notwendigen Merkmalsinformationen (z.B. Normalspannungen der Stäbe) reduziert werden können. Dazu ist angedacht, zukünftig nur noch eine durch den Grad der statischen Unbestimmtheit gegebene Anzahl zusätzlicher Merkmalsinformationen für den Eingabevektor der Trainingsdaten einzusetzen. So ließen sich Umfang der Datensysteme und Netztopologien sowie Aufwendungen für Simulation und VLSI-Design signifikant reduzieren. Aufbauend auf der vorgestellten Grundidee sind Echtzeit-Überwachungssysteme auf Basis trainierter Neuronaler Netze oder entsprechender VLSI-Schaltungen denkbar. Diese ordnen signifikante Veränderungen im Spektrum der Systemreaktionen einer belasteten Struktur aufgetretenen Schädigungen als Ursache zu. Die Systeme könnten fortlaufend mit definierten, aktuellen Informationen aus dem Betrieb des überwachten Systems aktualisiert werden. Die Vorteile dieser Echtzeit-Schadensanalyse ohne gleichzeitige Vergrößerung des Gefährdungspotentials wären:

- Erhöhung zulässiger Belastungen oder Reduktion des Strukturgewichts bei gleichbleibender Belastung
- Reduktion hoher Inspektionskosten durch Vergrößerung der Inspektionsintervalle
- Meldung aufgetretener Schädigungen in Echtzeit noch während des Betriebs

Die in Abschnitt 8.3 vorgestellten Ergebnisse sind auszugsweise in HERRMANN UND STRENG [39, 40] veröffentlicht. Das folgende Kapitel untersucht Korrelationen in der Matrixendarstellung statisch belasteter Fachwerke sowie zugehöriger Neuronaler Netze und baut dazu auf dem bisher gewonnenen Verständnis zur Nutzung problemspezifischer Neuronaler Netze in der Strukturmechanik auf.

9 Analogie Struktur – Neuronale Netze

Die Korrelation zwischen der Steifigkeitsmatrix einer Fachwerkstruktur und der Netztopologie eines zugehörigen, problemspezifischen Neuronalen Netzes ist Gegenstand der folgenden Untersuchungen. Dabei soll festgestellt werden, welche Neuronen oder Teilbereiche des Neuronalen Netzes welche Teilbereiche der Steifigkeitsmatrix des untersuchten Fachwerks repräsentieren. Dies dient einem besseren Verständnis beim Einsatz Neuronaler Netze in der Strukturmechanik.

Die der rechnerischen Ermittlung von Systemreaktionen (z.B. Knotenverschiebungen) einer statisch belasteten Fachwerkstruktur zugrundeliegende Problemstellung (Theorie I. Ordnung) entspricht der Lösung eines inhomogenen Linearen Gleichungssystems (iLGS). Die in dieser Arbeit nach Netzplan und Neuronenfunktionalität neu entwickelte, problemspezifische Netztopologie ist in der Lage, die Systemreaktionen einer Fachwerkstruktur (z.B. Knotenverschiebungen) für den jeweils aktuellen statischen Lastfall zu bestimmen. Die Vorgehensweise ist in vier Schritte aufgeteilt:

- Entwicklung von Netzplan und Neuronenfunktionalität einer problemspezifischen Netztopologie zur Lösung eines iLGS
- Test des neu entwickelten Lösungsverfahrens an einem mathematischen iLGS
- Herstellung einer Korrelation zwischen problemspezifischer Netztopologie und iLGS
- Test der Methodik an statisch bestimmten und unbestimmten Fachwerkstrukturen

Eine Diskussion der Vorgehensweisen und Ergebnisse beendet das Kapitel.

9.1 Neue Systematik

Die Entwicklung von Netzplan und Neuronenfunktionalität eines problemspezifischen Neuronalen Netzes (iLGS-Netztopologie) zur Abbildung eines iLGS der Form

$$\mathbf{X} \mathbf{a} = \mathbf{b} \quad (136)$$

und die anschließende Lösung des iLGS lassen sich in folgende Schritte gliedern:

- Identische Repräsentation des iLGS mit n Unbekannten in einer neu entwickelten problemspezifischen $(n - 1)$ -Netztopologie

- Generierung von n linear unabhängigen Trainingsmustern aus der Koeffizientenmatrix \mathbf{X} und dem Vektor der Absolutglieder \mathbf{b} des untersuchten iLGS
- Zufällige Initialisierung der Gewichtsmatrix \mathbf{W} und Training des Neuronalen Netzes auf einen gewünschten Gesamtnetzfehler E bzw. SSE

In dem iLGS in Gleichung (136) bezeichnet \mathbf{X} eine gegebene $(m \times n)$ -Koeffizientenmatrix, \mathbf{a} den $(n \times 1)$ -Vektor der gesuchten Lösung des iLGS und \mathbf{b} den gegebenen $(m \times 1)$ -Vektor der Absolutglieder. Die Gewichtsmatrix \mathbf{W} degeneriert für die eingesetzte $(n - 1)$ -Netztopologie zu einem n -dimensionalen Vektor \mathbf{w} . Die erfolgreich trainierten Verbindungsgewichte \mathbf{w} der entwickelten, problemspezifischen Netztopologie sind mit dem gesuchten Lösungsvektor \mathbf{a} des iLGS identisch.

Das Lösungsverhalten des iLGS in Gleichung (136) mit m Gleichungen und n Variablen ist durch den Rang $\text{Rg}(\mathbf{X})$ der Koeffizientenmatrix \mathbf{X} und den Rang $\text{Rg}(\mathbf{X}|\mathbf{b})$ der erweiterten Koeffizientenmatrix $(\mathbf{X}|\mathbf{b})$ bestimmt. Für $\mathbf{b} \neq \mathbf{0}$ ergibt sich aus

$$\text{Rg}(\mathbf{X}) = \text{Rg}(\mathbf{X}|\mathbf{b}) = n \quad (137)$$

die eindeutige Lösbarkeit des iLGS. Der Vektor der unbekanntenen Variablen \mathbf{a} ist damit eindeutig bestimmbar. Die m Zeilen $(x_{l1}, x_{l2}, \dots, x_{ln})$ der Koeffizientenmatrix \mathbf{X} mit $l = 1, \dots, m$ sind linear unabhängig falls \mathbf{X} den Rang $\text{Rg}(\mathbf{X}) = m$ hat. Für den Fall $m = n$ ist die Koeffizientenmatrix \mathbf{X} regulär, falls für die Determinante $\det(\mathbf{X})$ der Koeffizientenmatrix $\det(\mathbf{X}) \neq 0$ gilt. Auf diese Systeme und den quadratischen Fall $m = n$ seien die folgenden Untersuchungen beschränkt.

Ausgehend von dem allgemeinen Ansatz eines vollständig ebenenweise vernetzten Neuronalen Netzes in Abbildung 4 wird im folgenden eine Netztopologie vorgestellt, die in der Lage ist, ein iLGS der Form

$$\mathbf{X} \mathbf{w} = \mathbf{t} \quad (138)$$

identisch zu repräsentieren. Die nach Neuronenfunktionalität und Netzplan zu diesem Zweck neu entwickelte iLGS-Netztopologie ist in Abbildung 29 dargestellt. Die Transferfunktion des Ausgabeneurons basiert auf der Funktionalität des Standardneurons in Abbildung 3 (links) sowie der allgemeinen Transferfunktion aus Gleichung (67). Die Transferfunktion aller Eingabeneuronen i ist als Identität durch

$$f_{\text{trans},n i} = in_i \quad (139)$$

mit der Eingabe in_i des Trainingsmusters l mit $l = 1, \dots, p$ zum Eingabeneuron i für alle $i = 1, \dots, n$ gegeben. Für die Transferfunktion des Ausgabeneurons j gilt mit der Identität für

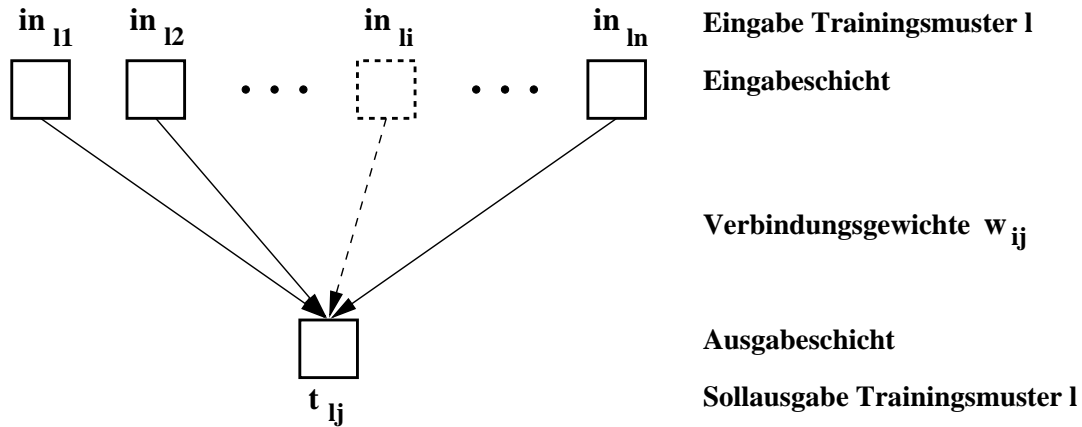


Abbildung 29: Problemspezifische Netztopologie zur Lösung eines iLGS

f_{akt} und f_{aus} sowie der Eingabefunktion $f_{ein} = \sum_{i=1}^n (w_{ij}(t) o_i)$ im Trainingszustand t die Formulierung

$$f_{trans,nj} = \sum_{i=1}^n w_{ij}(t) o_i . \quad (140)$$

Der Ausgabewert des Eingabeneurons i ist durch o_i gegeben. Damit folgt die Übertragungsfunktion des Netzwerks aus Abbildung 29 zum Trainingsmuster l und Ausgabeneuron j mit $j = n + 1$ zu

$$o_{lj}(t) = \sum_{i=1}^n w_{ij}(t) in_{li} . \quad (141)$$

Die Gesamtfehlerfunktion des Netzwerks in Abbildung 29 ist analog zu Gleichung (84) für ein Ausgabeneuron durch

$$E = \sum_{l=1}^p \frac{1}{2} (t_l - o_l(t))^2 \quad (142)$$

definiert. Nach erfolgreichem Training des iLGS-Netzwerks ergibt sich für den Gesamtfehler im Grenzfall $E = 0$. Mit $in_{li} = x_{li}$ folgt die Identität

$$t_l = o_l = \sum_{i=1}^n x_{li} w_{ij} \quad \text{für alle } l = 1, \dots, p . \quad (143)$$

Unter dieser Bedingung repräsentiert das Neuronale Netz aus Abbildung 29 (iLGS-Netztopologie) die Gleichung

$$\mathbf{t} = \mathbf{X} \mathbf{w} . \quad (144)$$

Die Korrelation zwischen der Netzübertragungsfunktion der iLGS-Netztopologie in Gleichung (144) und dem zu untersuchenden iLGS in Gleichung (136) wird durch die Systematik der ausgewählten Trainingsdaten hergestellt. Das Training der iLGS-Netztopologie erfolgt darauf aufbauend mit einer Kombination aus Elementen der Absolutglieder $\mathbf{b} =$

$(b_1, b_2, \dots, b_1, \dots, b_m)^T$ mit $l = 1, \dots, (p = m)$ und den Zeilenvektoren der Koeffizientenmatrix \mathbf{X} des zu lösenden iLGS aus Gleichung (136). Die Vektoren der Lernmuster $\mathbf{l}_l = (\mathbf{in}_l, \mathbf{t}_l)^T$ setzen sich aus den Zeilenvektoren von \mathbf{X} als Eingabevektoren $\mathbf{in}_l = (x_{l1}, x_{l2}, \dots, x_{ln})$ sowie dem zugehörigen Element l des Vektors \mathbf{b} der Absolutglieder als Sollausgabe $t_l = b_l$ zusammen. Es ergeben sich die Lernvektoren in der Form

$$\mathbf{l}_l = (x_{l1}, x_{l2}, \dots, x_{ln} | b_l)^T \text{ für alle } l = 1, \dots, m. \quad (145)$$

Ein Koeffizientenvergleich liefert die gesuchte Korrelation zwischen dem untersuchten iLGS aus Gleichung (136) in der Form

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad (146)$$

und der Abbildungsfunktion der problemspezifischen iLGS-Netztopologie in Gleichung (141) bzw. (143). Mit der Identität

$$\mathbf{X} \mathbf{a} = \mathbf{b} \Leftrightarrow \mathbf{X} \mathbf{w} = \mathbf{t} \quad (147)$$

folgt $\mathbf{a} = \mathbf{w}$.

Die hinsichtlich Neuronenfunktionalität, Netzplan und eingesetzten Trainingsdaten für die Lösung von iLGS neue Systematik ist damit in der Lage, den unbekanntem Lösungsvektor \mathbf{a} des iLGS $\mathbf{X} \mathbf{a} = \mathbf{b}$ über die Identität $\mathbf{a} = \mathbf{w}$ zu bestimmen. Die hiermit gefundene Korrelation zwischen iLGS und Netztopologie ordnet die Vektoren der Zeile l der Koeffizientenmatrix \mathbf{X} der Lerneingabe und damit den Neuronen der Eingabeschicht zu. Das zugehörige Element l der Absolutglieder wird der Sollausgabe der Trainingsmuster und damit dem Ausgabeneuron zugeordnet. Nach erfolgreichem Training mit einem beliebigen Lernalgorithmus besteht dann Gleichheit zwischen dem Vektor der trainierten Netzgewichte \mathbf{w} und dem Vektor der gesuchten Lösung \mathbf{a} des iLGS.

9.1.1 Mathematisches iLGS

Zur Veranschaulichung und Validierung der entwickelten Systematik wird ein einfaches mathematisches iLGS $\mathbf{X} \mathbf{a} = \mathbf{b}$ der Form

$$\begin{aligned} -1a_1 + 8a_2 + 3a_3 &= 2 \\ 2a_1 + 4a_2 - 1a_3 &= 1 \\ -2a_1 + 1a_2 + 2a_3 &= -1 \end{aligned} \quad (148)$$

untersucht. Die eindeutige Lösbarkeit des Gleichungssystems ergibt sich für $n = 3$ Unbekannte des Vektors \mathbf{a} aus $\text{Rg}(\mathbf{X}) = \text{Rg}(\mathbf{X}|\mathbf{b}) = 3$ und der Determinanten der Koeffizientenmatrix $\det(\mathbf{X}) = 5 \neq 0$. Die drei linear unabhängigen Muster der Trainingsdaten T_{18} entsprechen der in Abschnitt 9.1 beschriebenen Methodik und sind in Tabelle 30 gegeben. Zur Lösung des

Muster	Eingabe	Sollausgabe
1	-1 8 3	2
2	2 4 -1	1
3	-2 1 2	-1

Tabelle 30: Trainingsdaten T_{18} zu iLGS in Gleichung (148)

iLGS aus Gleichung (148) mit $n = 3$ Unbekannten ist gemäß der vorgestellten neuen Methodik eine $(3 - 1)$ -Netztopologie notwendig, die in Abbildung 30 dargestellt ist. Bei den getesteten

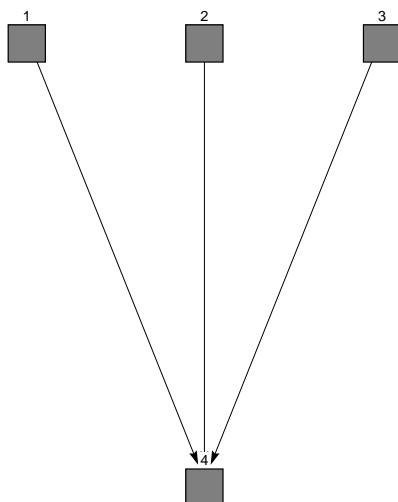


Abbildung 30: Netztopologie NT M zur Lösung des iLGS in Gleichung (148)

Lernverfahren handelt es sich um BackpropMomentum und Rprop. Die Verbindungsgewichte der Netztopologie werden vor dem Start des Lernverfahrens zufällig gleichverteilt aus dem Intervall $[-1.0, 1.0]$ initialisiert und die einzelnen Muster der Trainingsdaten T_{18} dem Netzwerk zu jeder trainierten Epoche in zufälliger Reihenfolge präsentiert. Die im folgenden vorgestellten Ergebnisse repräsentieren jeweils die besten Trainingsresultate zahlreicher Variationsrechnungen.

Das Lernverfahren BackpropMomentum liefert nach gleichverteilter Initialisierung der Verbindungsgewichte zu Beginn einen Netzfehler (siehe Gleichung (85)) von $rmSSE =$

766.55414. Nach 2600 Epochen ist $SSE < 10^{-5}$ und die Verbindungsgewichte zeigen die Werte $w_{14} = 5.00000$, $w_{24} = -1.00001$, $w_{34} = 5.00000$. Im Vergleich zur analytischen Lösung mit den Zielgewichten $z_{14} = 5$, $z_{24} = -1$, $z_{34} = 5$ ergibt sich bzgl. der externen Genauigkeit der Simulationssoftware SNNS ein individueller Approximationsfehler

$$AF_i = \left| \frac{z_{ij} - w_{ij}}{z_{ij}} \right| \quad (149)$$

von $AF_1 < 10^{-5} \%$, $AF_2 = 10^{-3} \%$, $AF_3 < 10^{-5} \%$. Rprop zeigt mit denselben Trainingsdaten T_{18} langsames Konvergenzverhalten und mit $AF_1 = 0.0134 \%$, $AF_2 = 0.017 \%$, $AF_3 = 0.0126 \%$ auch deutlich schlechtere Approximationsergebnisse.

Die Vorteile der inhärenten Parallelität Neuronaler Netze kommen erst bei einer Implementierung als parallele Simulation bzw. bei einer Implementierung als spezielle VLSI-Hardware zum tragen (siehe z.B. WANG [47]). Je nach Größe der Problemstellung bzw. Anzahl n der unbekannt Parameter des Vektors \mathbf{a} ist das vorgestellte, neu entwickelte Verfahren langsamer als spezielle mathematische Algorithmen. Ein Vergleich der vorgestellten Methodik mit mathematischen Verfahren zur Lösung von iLGS mit n Unbekannten hinsichtlich der dabei auftretenden Zeitkomplexität (siehe z.B. STOER [130]) zeigt folgende Ergebnisse:

- Der Gauß-Algorithmus als Standardreduktionsverfahren hat eine Zeitkomplexität der Ordnung $O(n^3)$.
- Für das Cholesky-Verfahren müssen $O(n)$ Quadratwurzeln berechnet und $O\left(\frac{n^3}{6}\right)$ Multiplikationen und Additionen ausgeführt werden.
- Aus der Regularität von \mathbf{X} folgt mit Gleichung (137) über die Cramersche Regel die Lösung des iLGS in Gleichung (136) zu $\mathbf{a} = \mathbf{X}^{-1} \mathbf{b}$. Der zur Bestimmung der Inversen \mathbf{X}^{-1} geeignete Gauß-Jordan-Algorithmus hat ebenfalls die Zeitkomplexität $O(n^3)$.
- Das Training einer *Feedforward*-Netztopologie für Ep Epochen bei m Trainingsmustern mit dem Lernalgorithmus BackpropMomentum hat die Zeitkomplexität $O(Ep \cdot m \cdot n)$. Damit ist die Effizienz des vorgestellten Lösungsverfahrens stark von der Anzahl der Lernepochen abhängig, die bis zum Erreichen eines gewünschten Fehlerniveaus benötigt werden.

9.1.2 Analogie Struktur – Problemspezifisches Neuronales Netz

Die gesuchte Korrelation zwischen der Steifigkeitsmatrix einer Fachwerkstruktur und der Netztopologie eines zugehörigen, problemspezifischen Neuronalen Netzes ergibt sich unter

den Annahmen der Theorie I. Ordnung aus der Darstellbarkeit der Zusammenhänge einer statisch belasteten Fachwerkstruktur mittels eines iLGS der Form

$$\mathbf{X} \mathbf{u} = \mathbf{f}. \quad (150)$$

Durch \mathbf{X} ist die $(m \times n)$ -Steifigkeitsmatrix der untersuchten Struktur gegeben. Der $(n \times 1)$ -Vektor \mathbf{u} bezeichnet die n unbekanntes Knotenverschiebungen des Systems und der $(m \times 1)$ -Vektor \mathbf{f} den aktuellen statischen Lastfall. Es gelte die Beschränkung auf quadratische Systeme durch $m = n$. Die verwendete Nomenklatur beschreibt das iLGS, die Trainingsdaten und die daraus resultierende Netztopologie nach Einstreichen der Randbedingungen in die Gesamtsteifigkeitsmatrix, die Lastspalte und den Vektor der unbekanntes Knotenverschiebungen.

Die gesuchte Zuordnung zwischen Netztopologie und Steifigkeitsmatrix ergibt sich aus einem Koeffizientenvergleich zwischen Gleichung (150) und Gleichung (144) zur Äquivalenz des Vektors \mathbf{u} der unbekanntes Systemreaktionen (Knotenverschiebungen) mit dem Vektor der trainierten Netzgewichte \mathbf{w} . Damit ist die Zuordnung der Neuronen der Ein- bzw. Ausgabeschicht zu einer definierten Position innerhalb der Steifigkeitsmatrix bzw. des Lastvektors analog zu den Ergebnissen von Abschnitt 9.1 gegeben. Die gefundene Korrelation ordnet die Vektoren der Zeile l der Koeffizientenmatrix \mathbf{X} (Gleichung (146)) $(x_{l1}, x_{l2}, \dots, x_{ln})$ mit $l = 1, \dots, m$ der Lerneingabe und damit in der Reihenfolge $1, \dots, n$ den Neuronen $1, \dots, n$ der Eingabeschicht zu. Das zugehörige Element l des Lastvektors \mathbf{f} wird der Sollausgabe der Trainingsmuster T und damit dem einzigen Ausgabeneuron zugewiesen.

Die Steifigkeitsmatrix einer Fachwerkstruktur ist in vielen Fällen nur entlang der beiden Hauptdiagonalen mit Werten ungleich null besetzt. Um das Konvergenzverhalten des Lernalgorithmus zu unterstützen, besteht die Möglichkeit, über eine bzgl. der Determinante der Steifigkeitsmatrix invariante Umformung die schwach besetzte Steifigkeitsmatrix in eine vollständig mit Werten ungleich null besetzte, modifizierte Steifigkeitsmatrix zu transformieren. Die orthogonale Householder-Matrix (siehe z.B. GILL ET AL. [131]) $\mathbf{H} = \mathbf{I} - \frac{1}{\beta} \boldsymbol{\omega} \boldsymbol{\omega}^T$ mit $\beta = \frac{1}{2} \|\boldsymbol{\omega}\|^2$, $\boldsymbol{\omega}$ als $(n \times 1)$ -Zufallsvektor und \mathbf{I} als Einheitsmatrix variiert das Originalgleichungssystem $\mathbf{X} \mathbf{u} = \mathbf{f}$ zu

$$\mathbf{H} \mathbf{X} \mathbf{H}^T \mathbf{H} \mathbf{u} = \mathbf{H} \mathbf{f} \Leftrightarrow \mathbf{X}_{\text{mod}} \mathbf{u}_{\text{mod}} = \mathbf{f}_{\text{mod}}. \quad (151)$$

Die so modifizierte Steifkeitsmatrix \mathbf{X}_{mod} ist vollständig belegt. Nach Bestimmung von \mathbf{u}_{mod} folgt die Lösung des Originalgleichungssystems \mathbf{u} aus der Rücktransformation $\mathbf{u} = \mathbf{H}^T \mathbf{u}_{\text{mod}}$. Dies erweist sich jedoch für die vorgeschlagene Systematik zur Lösung von iLGS mit Neuronalen Netzen als nicht notwendig, da bereits die schwach besetzte Steifigkeitsmatrix ausreichende Informationen zum Training des Neuronalen Netzes bereitstellt.

9.2 Statisch bestimmte Fachwerke

Die Lösung der iLGS zugehörig zu statisch bestimmten Beispielfachwerken mit 5 und 19 Stäben erfolgt durch Training problemspezifischer iLGS-Netztopologien. Die Untersuchungen dienen der Kontrolle der in Abschnitt 9.1 entwickelten Systematik und Bestätigung der gefundenen Korrelation.

Das in Abbildung 31 dargestellte ebene Fachwerk FW5 ist statisch bestimmt und in den Knoten 2 und 3 durch die statischen Lasten X_2, Y_2 und X_3, Y_3 belastet. Die Lagerung des Fachwerks unterbindet Verschiebungen des Knotens 1 in x- und y-Richtung sowie Verschiebungen des Knotens 4 in x-Richtung. Für den Vektor der Systemreaktionen $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5)^T$

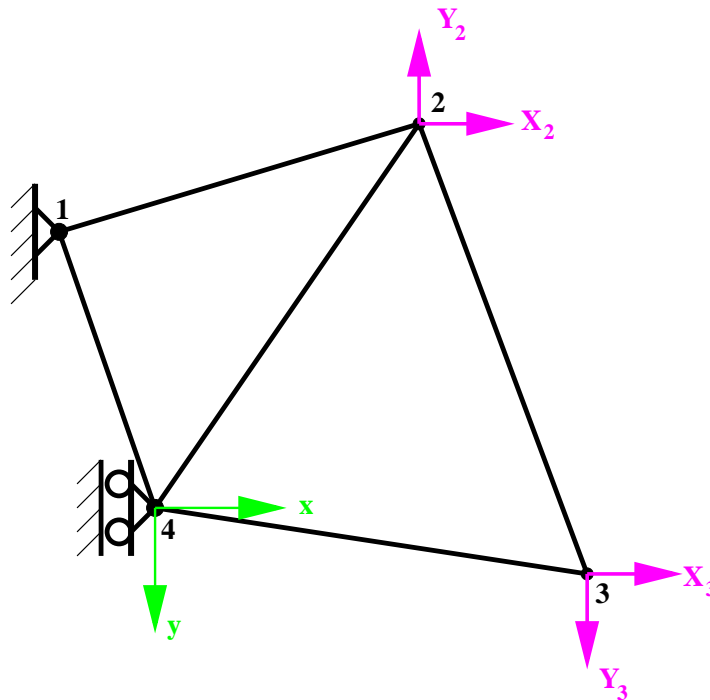


Abbildung 31: FW5: Statisch bestimmtes Fachwerk mit fünf Stäben

sind damit die Knotenverschiebungen $u_1 = \delta_{\text{Kn } 2,x}$, $u_2 = \delta_{\text{Kn } 2,y}$, $u_3 = \delta_{\text{Kn } 3,x}$, $u_4 = \delta_{\text{Kn } 3,y}$, $u_5 = \delta_{\text{Kn } 4,y}$ unbekannt. Die Trainingsdaten T_{19} zur Bestimmung der gesuchten Systemreaktion \mathbf{u} zu Fachwerk FW5 über die problemspezifische $(5 - 1)$ -Netztopologie NT N in Abbildung 32 entsprechen der durch Einstreichen der Randbedingungen reduzierten Gesamtsteifigkeitsmatrix sowie der zugehörigen Lastspalte gemäß der in Abschnitt 9.1 besprochenen Systematik.

Die Parameter der reduzierten Steifigkeitsmatrix und Lastspalte unterscheiden sich wegen der auftretenden, negativen und positiven Werte um bis zu sieben Größenordnungen. Dies führt

zu Konvergenzproblemen bis hin zur Divergenz des Lernverfahrens und damit zu schlechten Ergebnissen bei der Bestimmung der gesuchten Lösung. Ein Normierungsfaktor von 10^{-4} gleicht die Unterschiede äquivalent und ohne Informationsverlust aus. Die sich ergebenden Trainingsdaten T_{19} sind in Tabelle 31 zusammengestellt. Die Bestimmung der unbekannt

Muster	Eingabe					Sollausgabe
1	1.256771	-0.579471	-0.099982	-0.237457	0.148273	9.549297
2	-0.579471	1.484108	-0.237457	-0.563961	-0.518955	-6.366198
3	-0.099982	-0.237457	0.878950	0.562027	-0.324570	7.957747
4	-0.237457	-0.563960	0.562027	0.699198	-0.135237	6.366198
5	0.148273	-0.518955	-0.324570	-0.135237	1.240606	0.0

Tabelle 31: Trainingsdaten T_{19} zu Fachwerk FW5 in Abbildung 31

Knotenverschiebungen des Fachwerks FW5 ist äquivalent zur Lösung des iLGS der Form $\mathbf{X} \mathbf{u} = \mathbf{f}$ mit \mathbf{X} als reduzierte (5×5) -Steifigkeitsmatrix, \mathbf{u} als (5×1) -Vektor der unbekannt Knotenverschiebungen und \mathbf{f} als (5×1) -Vektor der statischen Belastung des Fachwerks. Die damit zur Lösung des iLGS benötigte $(5 - 1)$ -Netztopologie ist in Abbildung 32 dargestellt. Verbindungssystematik und Neuronenfunktionalität entsprechen den Angaben aus Abschnitt 9.1. Aufgrund der Resultate der Testrechnungen aus Abschnitt 9.1.1 wird beim

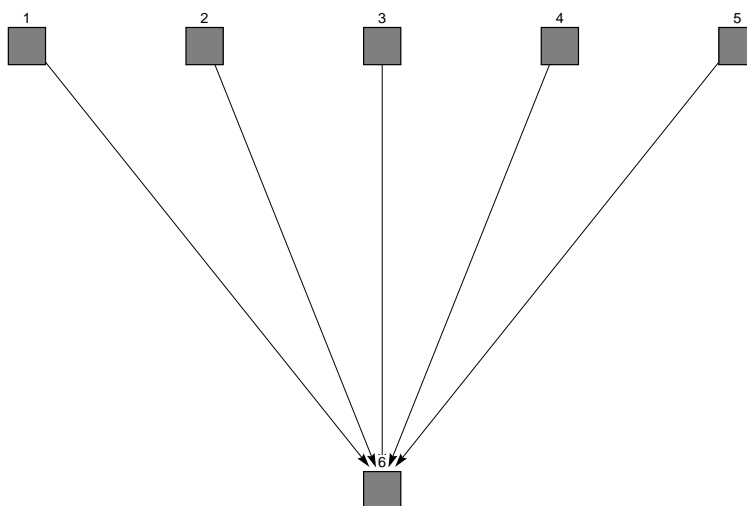


Abbildung 32: Netztopologie NT N zur Lösung des iLGS aus Fachwerk FW5

Training der problemspezifischen Netztopologie NT N nur der Lernalgorithmus Backprop-Momentum eingesetzt. Das Lernverfahren BackpropMomentum liefert nach gleichverteilter

Initialisierung der Verbindungsgewichte aus dem Intervall $[-1.0, 1.0]$ zu Beginn einen Netzfehler (gemäß Gleichung (85)) von $SSE = 219.85182$. Nach 400 Epochen ist $SSE < 10^{-5}$. Nachtraining um weitere 200 Epochen liefert die in Tabelle 32 gegebenen Verbindungsgewichte im Vergleich zur Lösung des entsprechend den Netztrainingsdaten aufgebauten iLGS mit MATLAB [132]. Weiteres Nachtraining der Netztopologie führt zu leichten Oszillatio-

Lösung	u_1	u_2	u_3	u_4	u_5
Netz	29.50416	30.50119	-8.58567	53.09888	12.77470
MATLAB	29.504157	30.501149	-8.585647	53.098830	12.774690

Tabelle 32: Vergleich der Lösungen des iLGS aus Fachwerk FW5

nen der Werte der Verbindungsgewichte ohne Verbesserung der Ergebnisse. Die individuellen Approximationsfehler relativ zu den mit MATLAB ermittelten Zielgewichten betragen $AF_1 = 8.41 \cdot 10^{-6} \%$, $AF_2 = 1.33 \cdot 10^{-4} \%$, $AF_3 = 2.59 \cdot 10^{-4} \%$, $AF_4 = 9.26 \cdot 10^{-5} \%$, $AF_5 = 7.24 \cdot 10^{-5} \%$.

Das in analoger Weise untersuchte iLGS eines statisch bestimmten und statisch belasteten Fachwerks mit 19 Stäben zeigt eine vergleichbar gute Approximation der Zielgewichte durch die Verbindungsgewichte des trainierten Netzwerks. Die auftretenden Approximationsfehler AF_i für alle $i = 1, \dots, 19$ nehmen dabei Werte kleiner oder gleich dem maximalen Fehlerniveau $AF_{13} = 5.14 \cdot 10^{-2} \%$ an. Der Approximationsfehler AF_{13} ist maximal, da das Zielgewicht $z_{13 \ 19}$ das betragsmäßig kleinste auftretende Zielgewicht für die gewählte Realisierung des Fachwerks ist.

Die Untersuchungen werden im folgenden Abschnitt auf statisch unbestimmte und dreidimensionale Fachwerke ausgedehnt.

9.3 Statisch unbestimmtes Fachwerk

Das Verhalten der vorgeschlagenen Methodik gegenüber statisch unbestimmten Fachwerken wird an einer dreidimensionalen Fachwerkstruktur untersucht. Die Fachwerkstruktur FW6 in Abbildung 33 ist einfach statisch unbestimmt und in den Knoten 1, 2 und 3 gelagert. Die Lagerung der Knoten 1 und 2 ist unverschieblich in x-, y- und z-Richtung, die Lagerung des Knotens 3 ist unverschieblich in z-Richtung und frei verschieblich in x- und y-Richtung. Die Belastung der Struktur erfolgt in Knoten 10 durch die statischen Lasten X_{10}, Y_{10}, Z_{10} entlang der Koordinatenachsen jeweils in positiver x-, y- und z-Richtung.

Die aus der Lagerung des Fachwerks resultierenden sieben Randbedingungen reduzieren die

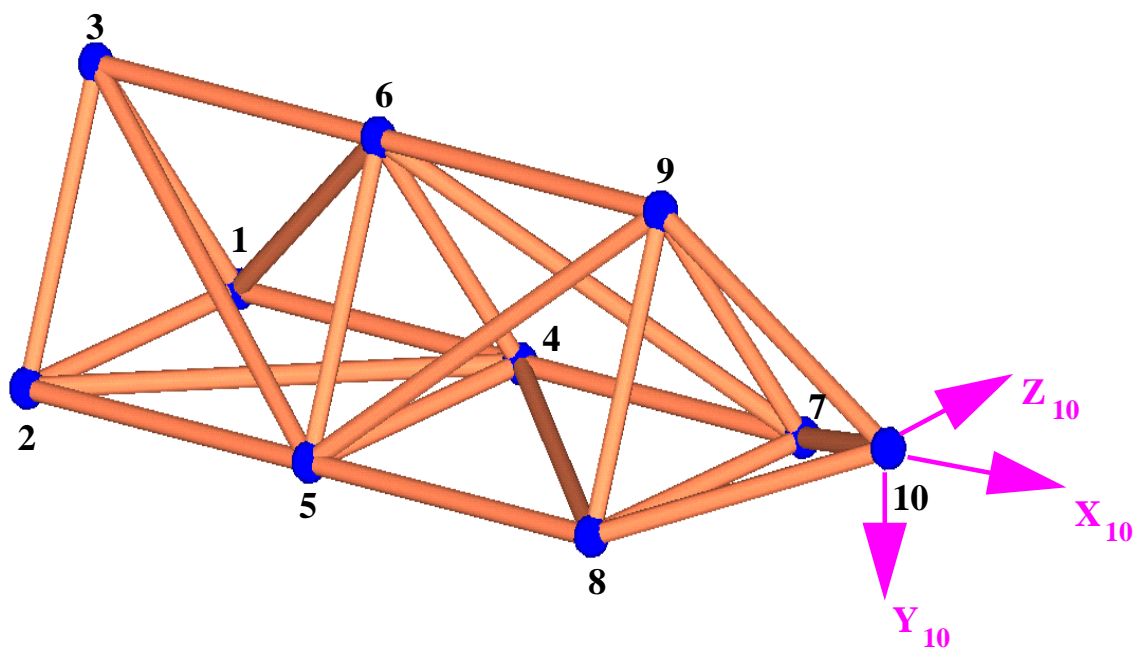


Abbildung 33: FW6: Statisch unbestimmtes Fachwerk mit 24 Stäben

Anzahl der unbekanntenen Knotenverschiebungen auf $n = 23$. Der (23×1) -Vektor \mathbf{u} der unbekanntenen Knotenverschiebungen ergibt sich aus der Lösung des iLGS $\mathbf{X} \mathbf{u} = \mathbf{f}$ mit \mathbf{X} als (23×23) -Steifigkeitsmatrix und \mathbf{f} als (23×1) -Lastvektor. Die zur Lösung des iLGS notwendige $(23 - 1)$ -Netztopologie ist in Abbildung 34 dargestellt. Die Trainingsdaten T_{20} sind

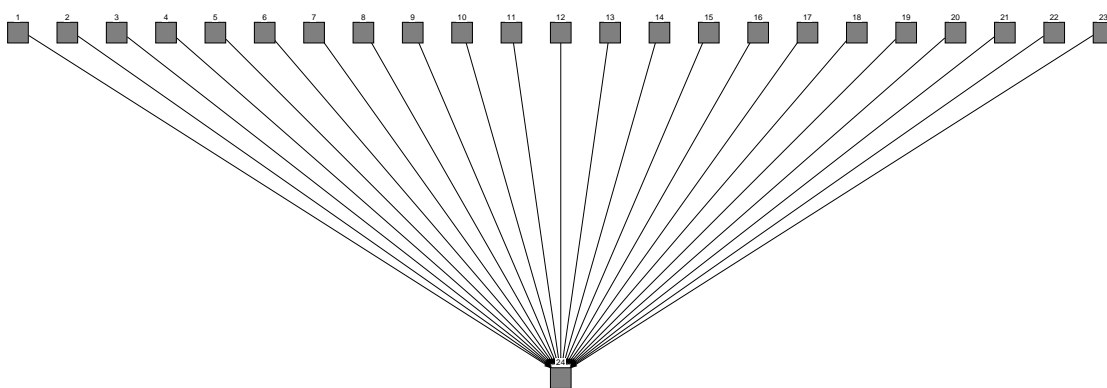


Abbildung 34: Netztopologie NT O zur Lösung iLGS Fachwerk FW6

aus Steifigkeitsmatrix und Lastvektor der untersuchten Fachwerkstruktur gemäß der Systematik aus Abschnitt 9.1 aufgebaut. Der Wertebereich der so entstehenden $m = 23$ Trainingsmuster umfaßt für die gewählte Realisierung des Fachwerks Werte, die bis zu zwölf

Größenordnungen auseinanderliegen. Die Normierungsfaktoren 10^{-4} bzw. 10^{-5} reduzieren den Wertebereich der normierten Trainingsdaten auf das Intervall $[-23.99, 52.68]$ für T_{20a} bzw. $[-2.399, 5.268]$ für T_{20b} . Damit verbessert sich das Konvergenzverhalten des Lernalgorithmus', und die Verwendung größerer Lernparameter $\eta, \alpha, c, d_{\max}$ (siehe Abschnitt 5.4.2) wird möglich. Dies führt zu einer Verminderung der Anzahl der notwendigen Trainingsepochen zur Erzielung ausreichender Konvergenz der Verbindungs- gegen die Zielgewichte. Gleichzeitig treten aus der Vergrößerung der Lernparameter aber vereinzelt Oszillationen des Gesamtnetzfehlers auf.

Der Lernalgorithmus ist BackpropMomentum. Die Steifigkeitsmatrix der gewählten Realisierung des Fachwerks FW6 ist schwach besetzt. Dennoch zeigt das Lernverfahren gute Konvergenz zu Fehlerniveaus von $SSE < 10^{-5}$. Die individuellen Approximationsfehler AF_i werden relativ zur numerischen Lösung des iLGS mit MATLAB basierend auf den Originaltrainingsdaten ermittelt. Die auftretenden Approximationsfehler AF_i für alle $i = 1, \dots, 23$ nehmen Werte kleiner oder gleich dem maximalen Fehlerniveau $AF_1 = 3.557 \cdot 10^{-2} \%$ für die Trainingsdaten T_{20a} und $AF_1 = 3.556 \cdot 10^{-2} \%$ für die Trainingsdaten T_{20b} an. Der Approximationsfehler AF_1 ist jeweils maximal, da das Zielgewicht $z_{1\ 24}$ das betragsmäßig kleinste auftretende Zielgewicht für die gewählte Realisierung des Fachwerks ist.

9.4 Fazit

Die entwickelte Systematik zur Lösung von iLGS mittels problemspezifischer Neuronaler Netze ist sowohl nach der Funktionalität der Neuronen als auch nach dem Netzplan der eingesetzten Netztopologie neu. Durch die Methode kann eine direkte Korrelation zwischen Neuronalen Netzen und den Steifigkeitsmatrizen der durch sie beschriebenen Fachwerkstrukturen hergestellt werden. Das Verfahren ist robust gegenüber schwach besetzten Koeffizientenmatrizen und Vektoren der Absolutglieder. Es bestehen keine Einschränkungen hinsichtlich dem Vorzeichen der Koeffizienten der zu lösenden iLGS.

Die entwickelte $(n - 1)$ -Netztopologie ist nach erfolgreichem Training in der Lage, den $(n \times 1)$ -Vektor der gesuchten Lösung des iLGS zu approximieren. Dies basiert auf der Neuronalen Netzen inhärenten Lernfähigkeit in Verbindung mit einer geeigneten Wahl von Neuronenfunktionalität und Netzplan als Approximation der gesuchten Lösung durch den Vektor der trainierten Verbindungsgewichte. Die Anzahl n der Unbekannten des iLGS ist stark beschränkt.

Die untersuchte Analogie zwischen Fachwerkstrukturen und einer zugehörigen Klasse problemspezifischer iLGS-Netztopologien ordnet die Zeilenvektoren der Steifigkeitsmatrix des

untersuchten Fachwerks der Lerneingabe und somit den Neuronen der Eingabeschicht zu. Das zugehörige Element des Lastvektors wird der Sollausgabe und damit dem Ausgabeneuron zugewiesen. Der Vektor der trainierten Netzgewichte \mathbf{w} approximiert über die in Abschnitt 9.1 hergeleitete Identität $\mathbf{w} = \mathbf{u}$ die unbekanntes Knotenverschiebungen der untersuchten Struktur.

Die Systematik ist in der Lage, n unbekanntes Systemreaktionen statisch bestimmter und unbestimmter Fachwerke zu ermitteln. Voraussetzungen sind die beschriebene Neuronenfunktionalität und der Netzplan der iLGS-Netztopologie zusammen mit einer erfolgreichen Minimierung des Gesamtnetzfehlers gegen null. Eine Skalierung der Trainingsdaten verbessert das Lern- und Konvergenzverhalten des Neuronalen Netzes deutlich bzw. ermöglicht sie überhaupt erst.

Das Konvergenzverhalten des Verfahrens und damit die erzielbare Approximationsgenauigkeit bei der Lösung von iLGS ist sehr sensitiv gegenüber der Wahl der Lernparameter des Trainingsverfahrens. Insgesamt liegen die Vorteile der neuen Methodik in folgenden Bereichen:

- Die vorgestellte Systematik ist für die Lösung mathematisch zu Gleichung (150) äquivalenter iLGS geeignet.
- Eine problemspezifische Netztopologie zur Lösung der iLGS ist für n unbekanntes Variablen einfach als $(n - 1)$ -Netztopologie (Eingabeschicht mit n Neuronen, keine verdeckte Schicht, Ausgabeschicht mit einem Neuron) anzulegen.
- Robustheit gegenüber schwach besetzten Koeffizientenmatrizen
- Inhärente Parallelität der Systematik
- Im Gegensatz zu dem Verfahren nach HERRMANN [10] und EMRICH [9] besteht in dem entwickelten Verfahren keine Beschränkung auf positive Koeffizienten von \mathbf{X} und \mathbf{b} .

Dem stehen folgende Nachteile gegenüber:

- Deutlich langsamer als spezialisierte Algorithmen (siehe Abschnitt 9.1.1)
- Konvergenz zum globalen Minimum ($E = 0$) auf der n -dimensionalen Hyperfläche des Netzfehlers in \mathbb{R}^{n+1} ist nicht garantiert
- Nur das iLGS zugehörig zur aktuellen Realisierung der Fachwerkstruktur und dem aktuellen Lastfall wird bzgl. der unbekanntes Knotenweggrößen gelöst.

Die vorgeschlagene Systematik ermittelt keine exakte Lösung des iLGS. Vielmehr wird die Matrizenmultiplikation aus Gleichung (150) als Regressionsproblem und damit als Minimierungsaufgabe der Gesamtfehlerfunktion E zugehörig zur Netzübertragungsfunktion der gewählten Netztopologie interpretiert. Über die Anpassung der Verbindungsgewichte durch den Lernalgorithmus entspricht dies einer Minimierung des Residuenvektors $\mathbf{d} = \mathbf{X} \mathbf{w} - \mathbf{t}$ der verwendeten euklidischen Metrik der Gesamtfehlerfunktion. Der Lernalgorithmus Backprop-Momentum basiert, wie in Abschnitt 5.4.2 beschrieben, nicht auf einer speziellen Numerik. Dies empfiehlt die vorgestellte Systematik besonders für schwach besetzte Koeffizientenmatrizen.

10 Zusammenfassung

Ziel der vorliegenden Arbeit ist die Integration des Lern- und Abstraktionspotentials geeigneter Neuronaler Netze in Systeme zur Optimierung und Überwachung technischer Strukturen. Zu diesem Zweck werden Möglichkeiten untersucht, Neuronale Netze zur Erfassung von Kriterien und Restriktionen einer Optimierungsaufgabe zu nutzen. Erfolgreich trainierte und gegebenenfalls durch Ausdünnungsverfahren (*Pruning*) reduzierte Neuronale Netze werden dann zusammen mit einer mehrgliedrigen Evolutionsstrategie in eine neuentwickelte Gesamtsystematik zur Lösung von Optimierungsaufgaben mit mehreren Optimierungszielen integriert.

Im Rahmen dieser Arbeit wird eine Systematik vorgeschlagen, die aufwendige und häufig auftretende Rechenoperationen für definierte Wertebereiche zu berücksichtigender Optimierungskriterien und -restriktionen durch trainierte Neuronale Netze ersetzt. Gleichzeitig wird die Leistungsfähigkeit Neuronaler Netze für ein Verfahren zur Überwachung und Lokalisierung von Schädigungen in Tragstrukturen ermittelt. Die Untersuchungen entwickeln eine Systematik zur Analyse unterschiedlicher Schädigungszustände am Beispiel statisch bestimmter und unbestimmter Fachwerkstrukturen nach Existenz, Ort und Ausmaß der Schädigungen. Das notwendige Verständnis zur Korrelation Neuronaler Netze mit den Steifigkeitsmatrizen der durch sie beschriebenen Tragwerkstrukturen ergibt sich aus Untersuchungen zur Lösung zugehöriger, inhomogener linearer Gleichungssysteme (iLGS) durch Training geeigneter Neuronaler Netze. Als zukünftiges Ziel ist hier z.B. an die Überwachung tragender Strukturen eines Flugzeugs in Echtzeit gedacht.

Die Implementierung von Funktionsansätzen in Neuronale Netze als Synthese aus geeignetem Netzplan und geeigneter Funktionalität der Neuronen erlaubt die Implementierung von bestehendem Grundwissen zur untersuchten Problemstellung in ein Neuronales Netz und damit die Generierung problemspezifischer Neuronaler Netze. Zusammen mit einer geeigneten Vorverarbeitung der Trainingsdaten (z.B. Dimensionsanalyse) können so physikalische Vorgänge über dem gesamten Gültigkeitsbereich des den Trainingsdaten zugrundeliegenden physikalischen Modells durch problemspezifische Neuronale Netze analytisch exakt erfaßt werden.

Eine Untersuchung multimodaler Optimierungsaufgaben mehrerer gleichzeitig zu berücksichtigender Optimierungskriterien und -restriktionen ist aufgrund der hierfür notwendigen Anzahl an Rechenoperationen äußerst aufwendig. Die zu dieser Problemstellung vorgeschlagene Systematik bindet die Erfassung von Kriterien und Restriktionen der gestellten Optimierungsaufgabe durch erfolgreich trainierte bzw. durch *Pruning* reduzierte konventionelle und problemspezifische Neuronale Netze in ein stochastisches Optimierungsverfahren (Mehrgliedri-

ge Evolutionsstrategie) zur Untersuchung der Optimierungsaufgabe ein. Das Vorgehen nutzt das Lernpotential und die schnellen Antwortzeiten trainierter Neuronaler Netze für die Untersuchung von Optimierungsaufgaben mit mehreren Optimierungszielen.

In der zur Schadensanalyse statisch bestimmter und unbestimmter Fachwerkstrukturen untersuchten Systematik werden Neuronale Netze zur Bestimmung des zugehörigen inversen Zusammenhangs verwendet. Die eingesetzten, ebenenweise vollständig vernetzten, konventionellen Ausgangsnetztopologien können neben reinem Training ohne Einbußen bei der erzielbaren Analysegenauigkeit signifikant durch das *Pruning*-Verfahren *Optimal Brain Surgeon* (OBS) hinsichtlich Anzahl an Verbindungen sowie Anzahl verdeckter Neuronen reduziert werden. Erfolgreich trainierte bzw. durch *Pruning* reduzierte Neuronale Netze sind in der Lage, unbekannte Schädigungen nach Existenz, Ort und Ausmaß einer oder mehrerer gleichzeitig auftretender Schädigungen für die berücksichtigte Schadensklasse des untersuchten Fachwerks zu klassifizieren. Die Erweiterung der Gesamtsystematik auf statisch unbestimmte Fachwerkstrukturen und der erfolgreiche Einsatz von *Pruning*-Verfahren zur Ausdünnung der Ausgangsnetztopologien sind neue Beiträge zu diesem Themenbereich.

Die der rechnerischen Ermittlung von Systemreaktionen (z.B. Knotenverschiebungen) einer statisch belasteten Fachwerkstruktur zugrundeliegende Problematik entspricht in der Strukturmechanik nach Theorie I. Ordnung der Lösung eines iLGS. Die in dieser Arbeit nach Netzplan und Neuronenfunktionalität neu entwickelte Netztopologie ist in der Lage, sämtliche Knotenverschiebungen der Fachwerkstruktur für den jeweils aktuellen statischen Lastfall zu ermitteln. Auf diese Weise kann die Steifigkeitsmatrix einer Fachwerkstruktur mit der Topologie des in der vorliegenden Arbeit entwickelten, problemspezifischen Neuronalen Netzes korreliert werden. Die vorgeschlagene Systematik interpretiert die Matrizenmultiplikation eines iLGS als Regressionsproblem und damit als Minimierungsaufgabe der Gesamtfehlerfunktion des zugehörigen Neuronalen Netzes. Die Methodik ist aufgrund der eingesetzten Lernalgorithmen auch für schwach besetzte Koeffizientenmatrizen der iLGS geeignet.

Die vorliegende Arbeit erweitert das Einsatzspektrum konventioneller und problemspezifischer Neuronaler Netze. Die selbstadaptive Ausdünnung Neuronaler Netze durch OBS erlaubt die problemgerechte Anpassung sehr allgemeiner Ausgangsnetztopologien und damit eine Minimierung der Trainings-, Speicher- und VLSI-Aufwendungen. Die gewonnenen neuen Erkenntnisse stellen einen Beitrag zum vertieften Verständnis der Leistungsfähigkeit Neuronaler Netze dar.

Literatur

- [1] MCAULAY, A.D.: *Optical Neural Network for Engineering Design*. NAECON, CH2596-5/88/0000-1302:1302–1306, 1988.
- [2] VANLUCHENE, R.D. und R. SUN: *Neural Networks in Structural Engineering*. Microcomputers in Civil Engineering, 5:207–215, 1990.
- [3] REHAK, D.R., C.R. THEWALT und L.B. DOO: *Neural Network Approaches in Structural Mechanics Computations*. In: NILSON, J.K. (Herausgeber): *Computer Utilization in Structural Engineering*, Seiten 168–176. ASCE Proceedings from Structure Congress, 1989.
- [4] SWIFT, R.A. und S.M. BATILL: *Application of Neural Networks to Preliminary Structural Design*. AIAA Journal, 1991.
- [5] PAO, Y.-H.: *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, USA, 1989.
- [6] GUNARATNAM, D.J. und J.S. GERO: *Effect of Representation on the Performance of Neural Networks in Structural Engineering Applications*. Microcomputers in Civil Engineering, 9:97–108, 1994.
- [7] HAJELA, P. und Z.P. SZEWCZYK: *Neurocomputing Strategies in Structural Design - on Analysing Weights of Feedforward Neural Networks*. Structural Optimization, 8:236–241, 1994.
- [8] BARTH, O.: *Funktionsapproximation mit Neuronalen Netzen*. Diplomarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart und IBK, DLR Stuttgart, DLR-IB 435-94/34 (1994), 1994.
- [9] EMRICH, G.: *Funktionsapproximation mit Neuronalen Netzen*. Studienarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart und IBK, DLR Stuttgart, DLR-IB 435-93/20 (1993), 1993.
- [10] HERRMANN, H.-G.: *Untersuchungen zur Anwendung Neuronaler Netze in der Strukturmechanik*. Doktorarbeit, Fakultät Luft- und Raumfahrttechnik der Universität Stuttgart, 1996.
- [11] RUDOLPH, S.: *Eine Methodik zur systematischen Bewertung von Konstruktionen*. Doktorarbeit, Fakultät Luft- und Raumfahrttechnik, Universität Stuttgart, 1994.

- [12] RUDOLPH, S.: *Entwurf, Anwendung und Interpretation Neuronaler Netze im Ingenieurwesen*. In: BERKHAHN, V. und M. OLBRICH (Herausgeber): *Forum Bauinformatik: Junge Wissenschaftler forschen*, Band 173. VDI Verlag, Düsseldorf, 1995.
- [13] RUDOLPH, S.: *On Topology and Generalization in Feed-Forward Neural Networks*. In: KRÖPLIN, B., S. RUDOLPH und G. EMRICH (Herausgeber): *Internationaler Workshop: Neuronale Netze in Ingenieurwissenschaften*, Universität Stuttgart, Februar 1996. Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen.
- [14] RUDOLPH, S.: *On Topology, Size and Generalization of Non-linear Feed-Forward Neural Networks*. *Neurocomputing*, 16(1):1–22, Juli 1997.
- [15] RUDOLPH, S.: *On a Data-Driven Model Identification Technique Using Artificial Neural Networks*. In: MOODY, J.E., S.J. HANSON und R.P. LIPPMANN (Herausgeber): *Proceedings EUROMECH 373, Colloquium on Modelling and Control of Adaptive Mechanical Structures*. Magdeburg, März 1998.
- [16] RUDOLPH, S. und B.-H. KRÖPLIN: *On Modelling Assumptions and Artificial Neural Networks*. *VDI Fortschrittsberichte*, 8(663):222–231, 1997.
- [17] RUDOLPH, S.: *On a Genetic Algorithm for the Selection of Optimally Generalizing Neural Network Topologies*. In: PARMEE, I.C. (Herausgeber): *Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control'96*, Seiten 79–86. University of Plymouth, Plymouth, United Kingdom, März 1996.
- [18] HAJELA, P. und L. BERKE: *Neurobiological Computational Models in Structural Analysis and Design*. In: *A Collection of Technical Papers I*, Band AIAA-90-1133-CP der Reihe *31st Structures, Structural Dynamics and Materials Conference*, Seiten 345–355. AIAA/ASME/ASCE/AHS/ASC, 1990.
- [19] BERKE, L. und P. HAJELA: *Application of Neural Nets in Structural Optimization*. In: ROZVANY, G.I.N. (Herausgeber): *Optimization of Large Structural Systems*, Band 231, I + II der Reihe *NATO ASI Series E*, Seiten 731–745, Kluwer Academic Publishers, Dordrecht, Holland, 1993. NATO Scientific Affairs Division.
- [20] BERKE, L. und P. HAJELA: *Application of Artificial Neural Nets in Structural Mechanics*. *Structural Optimization*, 4:90–98, 1992.
- [21] HAJELA, P. und L. BERKE: *Neural Network Based Decomposition in Optimal Structural Synthesis*. *Computing Systems in Engineering*, 2(5/6):473–481, 1991.

- [22] SZEWCZYK, Z. und P. HAJELA: *Neural Network Approximations in a Simulated Annealing Based Structural Design*. Structural Optimization, 5:159–165, 1993.
- [23] ZELL, A.: *Simulation Neuronaler Netze*. Addison-Wesley Publishing Company, Bonn, 1994.
- [24] SZEWCZYK, Z.P. und P. HAJELA: *Neurocomputing Strategies in Structural Design - Decomposition Based Optimization*. Structural Optimization, 8:242–250, 1994.
- [25] CARPENTER, W.C. und J.-F.M. BARTHELEMY: *A Comparison of Polynomial Approximations and Artificial Neural Nets as Response Surface*. Structural Optimization, 5:166–174, 1993.
- [26] SYLVESTER, J. und G. UHLMANN: *A Global Uniqueness Theorem for an Inverse Boundary Value Problem*. Annals of Mathematics, 125:153–169, 1987.
- [27] KAWIECKI, G.: *Application of Neural Networks to Damage Detection in Turbine Blades*. In: CHANG, FU-KUO [41], Seiten 456–465.
- [28] TSOU, P. und M.-H. HERMAN SHEN: *Structural Damage Detection and Identification Using Neural Networks*. AIAA Journal, 32(1), Januar 1994.
- [29] KUDVA, J.N., N. MUNIR und P.W. TAN: *Damage Detection in Smart Structures Using Neural Networks and Finite-Element Analysis*. Smart Materials and Structures, 1:108–112, 1992.
- [30] ELKORDY, M.F. und K.C. CHANG: *Neural Networks Trained by Analytically Simulated Damage States*. Journal of Computing in Civil Engineering, 7(2):130–145, April 1993.
- [31] ELKORDY, M.F., K.C. CHANG und G.C. LEE: *A Structural Damage Neural Network Monitoring System*. Microcomputers in Civil Engineering, 9(2):83–96, 1994.
- [32] RHIM, J. und S.W. LEE: *A Neural Network Approach for Damage Detection and Identification of Structures*. AIAA/ASME Adaptive Structures Forum, Seiten 173–180, 1994.
- [33] WORDEN, K.: *Structural Fault Detection Using a Novelty Measure*. Journal of Sound and Vibration, 201(1):85–111, 1997.
- [34] WORDEN, K., S.G. PIERCE, W.J. STASZEWSKI, W.R. PHILP, G.R. TOMLINSON und B. CULSHAW: *Detection of Faults in Carbon Fibre Reinforced Plates Using Lamb*

- Wave Based Novelty Detection*. In: *International Conference on Modern Practice in Stress and Vibration Analysis*, University College of Dublin, Ireland, September 1997.
- [35] WU, X., J. GHABOUSSI und J.H. GARRET JR.: *Use of Neural Networks in Detection of Structural Damage*. *Computers & Structures*, 42(4):649–659, 1992.
- [36] HERRMANN, H.-G.: *Schadenslokalisierung in Tragstrukturen mit Neuronalen Netzen*. Institutsbericht DLR-IB 435-95/6, Institut für Bauweisen- und Konstruktionsforschung, DLR Stuttgart, Februar 1995.
- [37] HERRMANN, H.-G. und K. STELLBRINK: *Schadenslokalisierung in Tragstrukturen mit Neuronalen Netzwerken*. *Zeitschrift für Flugwissenschaften und Weltraumforschung*, 20(3):163–168, Juni 1996.
- [38] HERRMANN, H.-G., C. BOLLER und K. STELLBRINK: *Some Aspects on the Use of Neural Networks in Damage Monitoring of Engineering Structures*. In: *DAMAS 97: International Workshop on Structural Damage Assessment Using Advanced Signal Processing Procedures*, University of Sheffield, UK, Juni/Juli 1997. EUROMECH 365.
- [39] HERRMANN, H.-G. und J. STRENG: *Problem-Specific Neural Networks for Detecting Structural Damage*. In: CHANG, FU-KUO [41], Seiten 267–278.
- [40] HERRMANN, H.-G. und J. STRENG: *Problem-Specific Neural Networks for Detecting Structural Damage*. In: VISCONTI, I. CRIVELLI (Herausgeber): *ECCM-8: European Conference on Composite Materials*, Band 3 der Reihe *Science, Technologies and Applications: Neapel, Italien, 3.-6. Juni*, Seiten 325–326. Woodhead Publishing Ltd., 1998.
- [41] CHANG, FU-KUO (Herausgeber): *Structural Health Monitoring: Current Status and Perspectives*, Band I der Reihe *Proceedings of the International Workshop on Structural Health Monitoring, Stanford University, September 18-20, 1997*. Technomic Publishing, Lancaster, PA, USA, 1997.
- [42] CHAKRABORTY, K., K. MEHROTRA, C.K. MOHAN und S. RANKA: *An Optimization Network for Solving a Set of Simultaneous Linear Equations*. In: *IJCNN International Joint Conference on Neural Networks*, Band 4, Seiten 516–521, New York, USA, 1992. IEEE.
- [43] FORBES, A.B. und A.J. MANSFIELD: *Neural Implementation of a Method for Solving Systems of Linear Algebraic Equations*. Technischer Bericht NPL Report DITC 155/89, National Physical Laboratory, 1989.

- [44] MARGARITIS, K.G., M. ADAMOPOULOS, M. LOUKAKIS und D.J. EVANS: *On the Solution of General Linear Systems and General Matrix Inversion Using Neural Network Architectures*. In: REIJNS, G.L. und J. LUO (Herausgeber): *Transputing in Numerical and Neural Network Applications*, Seiten 231–243. IOS Press, 1992.
- [45] POLYCARPOU, M.M. und P.A. IOANNOU: *Learning and Convergence Analysis of Neural-Type Structured Networks*. IEEE Transactions on Neural Networks, 3(1):39–50, Januar 1992.
- [46] REKANOS, I.T. und T.D. TSIBOUKIS: *A Parallel Solution Method for the Finite Elements Based on Hopfield Neural Networks. An Application to an Eddy Current Problem*. In: *Proceedings of the Seventh Biennial IEEE Conference on Electromagnetic Field Computation, CEFC*, Seite 242. IEEE, New York, 1996.
- [47] WANG, J.: *Electronic Realisation of Recurrent Neural Network for Solving Simultaneous Linear Equations*. Electronics Letter, 28(5):493–495, Februar 1992.
- [48] WANG, L. und J.M. MENDEL: *Structured Trainable Networks for Matrix Algebra*. In: *IJCNN International Joint Conference on Neural Networks*, Band 2, Seiten II–125 II–132, New York, USA, 1990. IEEE.
- [49] WANG, L.-X. und J.M. MENDEL: *Parallel Structured Networks for Solving a Wide Variety of Matrix Algebra Problems*. Journal of Parallel and Distributed Computing, 14:236–247, 1992.
- [50] WANG, L. und J.M. MENDEL: *Three-Dimensional Structured Networks for Matrix Equation Solving*. In: JUANG, B.H., S.Y. KUNG und C.A. KAMM (Herausgeber): *Neural Networks for Signal Processing*, Proceedings of the 1991 IEEE Workshop, Seiten 80–89, New York, USA, 1991. IEEE.
- [51] GÖPFERT, A., L. BITTNER, K.-H. ELSTER, F. NOŽIČKA, J. PIEHLER und R. TICHATSCHKE: *Lexikon der Optimierung*. Akademie Verlag, Berlin, 1986.
- [52] SCHWEFEL, H.-P.: *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser Verlag, Basel, 1977.
- [53] SCHITTKOWSKI, K., C. ZILLOBER und R. ZOTEMANTEL: *Numerical Comparison of Nonlinear Programming Algorithms for Structural Optimization*. Structural Optimization, 7:1–19, 1994.

- [54] ARORA, J.S., O.A. ELWAKEIL, A.I. CHAHANDE und C.C. HSIEH: *Global Optimization Methods for Engineering Applications: A Review*. Structural Optimization, 9:137–159, 1995.
- [55] SPELLUCCI, P.: *Numerische Verfahren der nichtlinearen Optimierung*. Birkhäuser-Verlag, Basel, 1993.
- [56] VIDAL, R.V.V. (Herausgeber): *Applied Simulated Annealing*, Band 396 der Reihe *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, 1993.
- [57] DUECK, G.: *New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel*. Journal of Computational Physics, 104:86–92, 1993.
- [58] DUECK, G. und T. SCHEUER: *Threshold Accepting: A General Purpose Optimization Algorithm Superior to Simulated Annealing*. Journal of Computational Physics, 90:161–175, 1990.
- [59] GÖRTLER, H.: *Dimensionsanalyse*. Springer Verlag, Berlin, 1975.
- [60] BRIDGMAN, P.W.: *Theorie der physikalischen Dimensionen*. B.G. Teubner, Leipzig und Berlin, 1932.
- [61] LANGHAAR, H.L.: *Dimensional Analysis and Theory of Models*. John Wiley & Sons, Inc., New York, 1951.
- [62] PETZOLD, TH.: *Ein Beitrag zur Entwurfsoptimierung ebener Fachwerke mit Evolutionsstrategien*. Doktorarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Fakultät Luft- und Raumfahrttechnik, Universität Stuttgart, 1998.
- [63] SCHMIT, L.A.: *Structural Optimization - Some Key Ideas and Insights*. In: ATREK, E., R.H. GALLAGHER, K.M. RAGSDALL und O.C. ZIENKIEWICZ (Herausgeber): *New Directions in Optimum Structural Design*, Seiten 1–45. John Wiley & Sons, Inc, 1984.
- [64] PARETO, V.: *Manual of Political Economy*. The Macmillan Press Ltd., London, Übersetzung der französischen Ausgabe von 1927 von A.S. Schwier, 1971.
- [65] SALUKVADZE, M.E.: *On the Existence of Solutions in Problems of Optimization under Vector-Valued Criteria*. Journal of Optimization Theory and Applications, 13(2), 1974.
- [66] SATTLER, H.-J.: *Ersatzprobleme für Vektoroptimierungsaufgaben und ihre Anwendung in der Strukturmechanik*. Technischer Bericht 88, Fortschrittsbericht der VDI-Zeitschriften, 1982.

- [67] PESCHEL, M.: *Ingenieurtechnische Entscheidungen*. VEB Verlag Technik, Berlin, 1980.
- [68] KREISSELMEIER, G. und R. STEINHAUSER: *Systematic Control Design by Optimizing a Vector Performance Index*. In: *IFAC Symposium, Computer Aided Design on Control System*. Zürich, Schweiz, August 1979.
- [69] MLEJNEK, H.P.: *Some Explorations in the Genesis of Structures*. In: BENDSOE, M.P. und C.A. MOTA SOARES (Herausgeber): *Topology Design of Structures*, Band 227, Serie E der Reihe *NATO Advanced Science Institute Series: Applied Sciences*, Seiten 287–300. Kluwer Academic Publishers, 1992.
- [70] STADLER, W.: *Multicriteria Optimization in Engineering and in the Sciences*, Band 37 der Reihe *Mathematical Concepts and Methods in Science and Engineering*. Plenum Press, New-York, USA, 1988.
- [71] GÖPFERT, A. und R. NEHSE: *Vektoroptimierung*, Band 74. B.G. Teubner Verlagsgesellschaft, Leipzig, 1990.
- [72] ESCHENAUER, H., J. KOSKI und A. OSYCZKA: *Multicriteria Design Optimization*. Springer Verlag, Berlin, 1990.
- [73] KURSAWE, F.: *Evolutionsstrategien für die Vektoroptimierung*. Diplomarbeit, Universität Dortmund, Fachbereich Informatik, Dortmund, 1990.
- [74] KURSAWE, F.: *A Variant of Evolution Strategies for Vector Optimization*. In: SCHWEFEL, H.-P. und R. MÄNNER (Herausgeber): *Parallel Problem Solving from Nature I*, Lecture Notes in Computer Science, Seiten 193–197, Berlin, 1990. Dortmund, Deutschland, 1.-3. Oktober, 1990, Springer Verlag.
- [75] RECHENBERG, I.: *Evolutionsstrategien*. In: *Simulationsmethoden in der Medizin und Biologie*, Berlin, 1978. Workshop Hannover 1977, Springer Verlag.
- [76] STRENG, J.: *Anpassung von Evolutionsstrategien bei der Lösung technischer Optimierungsprobleme*. Diplomarbeit, Institut für Flugzeugbau, Fakultät Luft- und Raumfahrttechnik, Universität Stuttgart und Institut für Bauweisen- und Konstruktionsforschung, DLR Stuttgart, DLR-IB 435-94/1, März 1994.
- [77] SCHWEFEL, H.-P.: *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. John Wiley & Sons, 1995.

- [78] BÄCK, TH.: *Evolutionary Algorithms in Theory and Practice*. Doktorarbeit, Universität Dortmund, Fachbereich Informatik, Februar 1994.
- [79] HAN, S.-P.: *A Globally Convergent Method for Nonlinear Programming*. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.
- [80] POWELL, M.J.D.: *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*. In: WATSON, G.A. (Herausgeber): *Numerical Analysis, Lecture Notes in Mathematics*, Band 630. Springer Verlag, Berlin, 1977.
- [81] SCHITTKOWSKI, K.: *On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function*. *Mathematische Operationsforschung und Statistik: Series Optimization*, 14(2):197–216, 1983.
- [82] FLEURY, C. und V. BRAIBANT: *Structural Optimization - A New Dual Method Using Mixed Variables*. *International Journal for Numerical Methods in Engineering*, 23:409–428, 1986.
- [83] SVANBERG, K.: *The Method of Moving Asymptotes - A New Method for Structural Optimization*. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
- [84] SVANBERG, K.: *A Globally Convergent Version of MMA without Linesearch*. In: ROZVANY, G.I.N. und N. OLHOFF (Herausgeber): *First World Congress of Structural and Multidisciplinary Optimization*, Band WCSMO-1 der Reihe *International Society for Structural and Multidisciplinary Optimization ISSMO*, Seiten 9–16. Elsevier Science, Pergamon, 1995.
- [85] RINNOOY KAN, A.H.G und G.T. TIMMER: *Global Optimization*. In: G.L. NEMHAUSER, ET AL. (Herausgeber): *Handbook in Operations Research and Management Science*, Band 1, Seiten 631–662. Elsevier Science Publishers, North-Holland, 1989.
- [86] GRUNDMANN, H.: *Zuverlässigkeit der Bauwerke*. In: *Berichte zur Zuverlässigkeitstheorie der Bauwerke, Abschlußbericht Sonderforschungsbereich 96*. Laboratorium für den Konstruktiven Ingenieurbau (LKI), TU München, Bericht (86/1989), 1989.
- [87] ABDO SARRAS, T.: *Zur Zuverlässigkeitsrechnung von statisch beanspruchten Tragwerken mit unsicheren Systemeigenschaften*. In: *Berichte zur Zuverlässigkeitstheorie der Bauwerke*. Laboratorium für den Konstruktiven Ingenieurbau (LKI), TU München, Bericht (88/1990), 1990.

- [88] HARTUNG, J., B. ELPELT und K.-H. KLÖSENER: *Statistik*. R. Oldenbourg Verlag, München, 1993.
- [89] HOHENBICHLER, M. und R. RACKWITZ: *Non-Normal Dependent Vectors in Structural Safety*. Journal of the Engineering Mechanics Division, 107(EM6):1227–1238, 1981.
- [90] GOLLWITZER, S.: *COMREL and SYSREL: Users Manual Version 7*. RCP Consult GmbH, www.strurel.de, München, Januar 1998.
- [91] HASOFER, A.M. und N.C. LIND: *Exact and Invariant Second-Moment Code Format*. Journal of the Engineering Mechanics Division, 100(EM1):111–121, Februar 1974.
- [92] RACKWITZ, R.: *Practical Probabilistic Approach to Design*. CEB Bulletin d'Information, 112:13–71, 1976.
- [93] ROSENBLATT, M.: *Remarks on a Multivariate Transformation*. Annals of Mathematical Statistics, 23:470–471, 1952.
- [94] RACKWITZ, R. und B. FIESSLER: *Structural Reliability under Combined Random Load Sequences*. Computers & Structures, 9:489–494, 1978.
- [95] MADSEN, H.O., S. KRENK und N.C. LIND: *Methods of Structural Safety*. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1986.
- [96] ADELI, H.: *Advances in Design Optimization*. Chapman & Hall, London, UK, 1994.
- [97] SCHUËLLER, G.I.: *Einführung in die Sicherheit und Zuverlässigkeit von Tragwerken*. Verlag Wilhelm Ernst & Sohn, Berlin, 1981.
- [98] THOFT-CHRISTENSEN, P. und M.J. BAKER: *Structural Reliability Theory and its Applications*. Springer Verlag, Berlin, 1982.
- [99] SPAETHE, G.: *Die Sicherheit tragender Konstruktionen*. Springer Verlag, Wien, 1992.
- [100] GOLLWITZER, S., T. ABDO SARRAS und R. RACKWITZ: *FORM - First Order Reliability Method*. RCP GmbH, München, November 1990.
- [101] KANDEL, E.R., J.H. SCHWARTZ und T.M. JESSELL: *Principles of Neural Science*. Prentice Hall International Inc., London, UK, 3. Auflage, 1995.
- [102] ROJAS, R.: *Neural Networks: A Systematic Introduction*. Springer Verlag, Berlin, 1996.

- [103] HASSOUN, M.H.: *Fundamentals of Artificial Neural Networks*. A Bradford Book, MIT Press, Cambridge, Massachusetts, USA, 1995.
- [104] ZELL, A., G. MAMIER, M. VOGT und N. MACHE ET AL.: *Stuttgart Neural Network Simulator, Version 4.1*. Report 6/95, Institute for Parallel and Distributed High Performance Systems (IPVR), Applied Computer Science – Image Understanding, Universität Stuttgart, 1995.
- [105] V.VYŠNIAUSKAS, F.C.A. GROEN und B.J.A. KRÖSE: *The Optimal Number of Learning Samples and Hidden Units in Function Approximation with a Feedforward Network*. Technical Report: CS-93-15, University of Amsterdam, Faculty of Computer Science and Mathematics, Amsterdam, Holland, 1993.
- [106] HORNIK, K. und M. STINCHCOMBE: *Multilayer Feedforward Networks are Universal Approximators*. In: WHITE, H. (Herausgeber): *Artificial Neural Networks*, Seiten 12–28. Blackwell Publishers, Cambridge, MA, USA, 1992.
- [107] STINCHCOMBE, M.: *Universal Approximation Using Feedforward Networks with Non-sigmoid Hidden Layer Activation Functions*. In: WHITE, H. (Herausgeber): *Artificial Neural Networks*, Seiten 29–40. Blackwell Publishers, Cambridge, MA, USA, 1992.
- [108] STINCHCOMBE, M.: *Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights*. In: WHITE, H. (Herausgeber): *Artificial Neural Networks*, Seiten 41–54. Blackwell Publishers, Cambridge, MA, USA, 1992.
- [109] CHEN, T., H. CHEN und R. LIU: *Approximation Capability in $C(\widetilde{\mathbb{R}}^n)$ by Multilayer Feedforward Networks and Related Problems*. IEEE Transactions on Neural Networks, 6(1):25–30, January 1995.
- [110] HERRMANN, H.-G.: *On Functional Analysis with Neural Networks*. Smart Materials and Structures, 7(1):52–62, February 1998.
- [111] AHRENS, H. und D. DINKLER: *Finite-Element-Methoden Teil I*. Institut für Statik der Technischen Universität Braunschweig, Bericht Nr. 88-50, 1994.
- [112] MCCLELLAND, J.L. und D.E. RUMELHART: *Explorations in Parallel Distributed Processing*. The MIT Press, Cambridge, Massachusetts, 1988.

- [113] RIEDMILLER, M.: *Untersuchung zu Konvergenz und Generalisierungsfähigkeit überwachter Lernverfahren mit dem SNNS*. In: ZELL, A. (Herausgeber): *Workshop SNNS-93: Simulation Neuronaler Netze mit SNNS*, Seiten 107–116. Universität Stuttgart, Fakultät Informatik: Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR), 1993. Bericht Nr. 10/93.
- [114] RIEDMILLER, M.: *Rprop - Description and Implementation Details*. Technischer Bericht 01/1994, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, 1994.
- [115] REED, R.: *Pruning Algorithms - A Survey*. IEEE Transactions on Neural Networks, 4(5):740–747, September 1993.
- [116] SCHREINER, T.: *Ausdünnungsverfahren für Neuronale Netze*. Diplomarbeit Nr. 1140, Fakultät Informatik, Universität Stuttgart, Juli 1994.
- [117] MORIARTY, D.E. und R. MIIKKULAINEN: *Forming Neural Networks through Efficient and Adaptive Coevolution*. Evolutionary Computation, 5(4):373–399, Ausgabe Winter 1997.
- [118] WHITLEY, W. und C. BOGART: *The Evolution of Connectivity: Pruning Neural Networks Using Genetic Algorithms*. In: *Proceedings of the Int. Joint Conference on Neural Networks*, Band I, Seiten 134–137, Washington, DC, USA, 1990.
- [119] GRÜNINGER, T.: *Optimierung rückkopplungsfreier Neuronaler Netze mittels Genetischer Algorithmen*. Studienarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, August 1995.
- [120] HASSIBI, B. und D.G. STORK: *Second Order Derivatives for Network Pruning: Optimal Brain Surgeon*. In: SEJNOWSKI, T.J., G.E. HINTON und D.S. TOURETZKY (Herausgeber): *Advances in Neural Information Processing Systems*, Band 5, Seiten 164–171. Morgan Kaufmann Publishers Inc., 1993.
- [121] HASSIBI, B., D. STORK und G. WOLFF: *Optimal Brain Surgeon and General Network Pruning*. In: *Proceedings on the IEEE International Conference on Neural Networks*, Band 1, Seiten 293–299. IEEE Press, New York, 1993.
- [122] HASSIBI, B., D.G. STORK, G. WOLFF und T. WATANABE: *Optimal Brain Surgeon: Extensions and Performance Comparisons*. In: COWAN, J.D., G. TESAURO und J. ALSPECTOR (Herausgeber): *Advances in Neural Information Processing Systems*, Band 6, Seiten 263–270. Morgan Kaufmann Publishers Inc., 1994.

- [123] CUN, Y. LE, J.S. DENKER und S.A. SOLLA: *Optimal Brain Damage*. In: TOURETZKY, D.S. (Herausgeber): *Advances in Neural Information Processing Systems*, Band 2, Seiten 598–605. Morgan Kaufmann Publishers Inc., 1990.
- [124] HERTZ, J., A. KROGH und R.G. PALMER: *Introduction to the Theory of Neural Computation*, Band I der Reihe *Lecture Notes of the Santa Fé Institute Studies in the Science of Complexity*. Addison-Wesley, Redwood City, CA, USA, 1991.
- [125] SCHÖNING, U.: *Theoretische Informatik - kurzgefaßt*. Spektrum Akademischer Verlag, Heidelberg, 1997.
- [126] NAAS, J. und H.L. SCHMID: *Mathematisches Wörterbuch mit Einbeziehung der theoretischen Physik*, Band I und II. Akademie Verlag GmbH, Berlin, B.G. Teubner Verlagsgesellschaft, Stuttgart, 2. Auflage, 1962.
- [127] EMRICH, G.: *Neuronale Bilderkennung mit neuen, multi-skaleninvarianten Zentralmomenten*. In: KRÖPLIN, B., S. RUDOLPH und G. EMRICH (Herausgeber): *Internationaler Workshop: Neuronale Netze in Ingenieurwissenschaften*, Universität Stuttgart, Februar 1996. Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen.
- [128] BERECHNUNGSUNTERLAGEN, INDUSTRIE-AUSSCHUSS STRUKTUR: *Luftfahrttechnisches Handbuch (LTH)*, Band G. IASB, 1988.
- [129] LOUIS, A.K.: *Inverse und schlecht gestellte Probleme*. Teubner Studienbücher: Mathematik, Stuttgart, 1989.
- [130] STOER, J.: *Einführung in die Numerische Mathematik I*. Springer Verlag, Berlin, 1979.
- [131] GILL, P.E., W. MURRAY und M.H. WRIGHT: *Practical Optimization*. System Optimization Laboratory, Department of Operations Research, Stanford University, CA, USA. Academic Press, Hartcourt Brace and Company, Publishers, San Diego, USA, 1981-1997.
- [132] MATLAB: *High-Performance Numeric Computation and Visualization Software*. The MathWorks, Inc., Natick, MA, USA, November 1996. Version 5.0.0.4064.