Evolutionäre Optimierung eines biologisch motivierten visuellen Objekterkennungssystems

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

der

Technischen Fakultät
der Universität Bielefeld

vorgelegt von

Georg Schneider
aus
St. Wendel

Oktober 2004

Danksagung

Angefertigt wurde diese Arbeit am Honda Research Institute Europe (HRI-EU) in Offenbach in der Arbeitsgruppe Evolutionary and Learning Technology (EL-Tec). Dem Leiter des Instituts Herrn Professor Dr. Edgar Körner danke ich sehr herzlich dafür, dass er mir ermöglicht hat, an dieser interessanten Fragestellung unter sehr förderlichen Bedingungen forschen zu können. Außerdem danke ich ihm für die konstruktive fachliche Unterstützung der Arbeit.

Mein ganz besonderer Dank gilt dem Leiter der EL-Tec Gruppe Herrn Dr. Bernhard Sendhoff für die Einführung in das vielschichtige und komplexe Feld der evolutionären Optimierung. Mit seinem umfangreichen Know-how hat er mir stets geholfen, diese Arbeit auf aussichtsreiche Wege zu führen.

Für die unermüdliche Unterstützung auf dem Gebiet der biologisch motivierten Objekterkennungssysteme bin ich Herrn Dr. Heiko Wersing besonders dankbar. Mit vielen wertvollen Anregungen und konstruktiver Kritik trug er wesentlich zum Gelingen dieser Arbeit bei.

Die kollegiale und angenehme Arbeitsatmosphäre am Honda Research Institute – auch über die Arbeit hinaus – weiß ich sehr zu schätzen. Meinen lieben Kolleginnen und Kollegen herzlichen Dank!

Die vorliegende Arbeit entstand als externe Dissertation in der Arbeitsgruppe Neuroinformatik der technischen Fakultät der Universität Bielefeld. Mein besonderer Dank gilt hier dem Leiter der Arbeitsgruppe Herrn Professor Dr. Ritter für die freundliche Betreuung dieser Arbeit und für wertvolle wissenschaftliche Anregungen.

Herrn Professor Dr. Gerhard Sagerer sage ich herzlichen Dank für die Bereitschaft, die Leitung der Prüfungskommission zu übernehmen. Ich bedanke mich auch bei Herrn Dr. Jochen Steil, der sich freundlicherweise bereit erklärt hat, Mitglied der Prüfungskommission zu sein.

Ingrid Konrad, Alexandra Mark, Dr. Stefan Menzel und meine Freundin Karin Runkel haben das Korrekturlesen übernommen: Herzlichen Dank!

Der menschliche Geist kommt zwar auf allerlei Erfindungen, aber er wird nie eine schönere, einfachere oder unmittelbarere Erfindung ersinnen als die Natur, da ihren Erfindungen nichts fehlt und nichts überflüssig ist.

Leonardo da Vinci

Inhaltsverzeichnis

Kapitel 1

Einleitung

Motivation

Der Mensch ist in der Lage, eine Vielzahl von einmal betrachteten Objekten wiederzuerkennen und voneinander zu unterscheiden. Diese Klassifikationsaufgabe gelingt ihm dabei unabhängig von zahlreichen Variationen der Objektansicht verursacht durch: wechselnde Beleuchtung, Rotation, Verschiebung (Translation) sowie teilweise Verdeckung der Objekte. Technische Systeme sind gegenwärtig noch weit von dieser menschlichen Leistungsfähigkeit entfernt.

Aufbauend auf visuellen Verarbeitungsprinzipien des Menschen wurde von Wersing und Körner ein visuelles Objekterkennungssystem [49, 50] vorgeschlagen, welches sich in eine größere, von Körner et al. skizzierte bidirektionale Verarbeitungsarchitektur [22] einordnen lässt. Dieses visuelle System basiert auf einer sequentiellen Abfolge von neuronalen Schichten zum Aufbau einer Hierarchie von optischen Merkmalen. Hierbei werden zu Beginn der Hierarchie zunächst einfache und dann darauf aufbauend zunehmend komplexere Merkmale verarbeitet. Das visuelle System ähnelt in seinem Aufbau dem von Fukushima entwickelten Neocognitron [8]. Ein weiteres Kennzeichen ist die Verwendung von einer Reihe von Nichtlinearitäten wie z.B. einer lateralen Inhibition zwischen parallel verlaufenden Merkmalspfaden. Trotz der Verwendung von neurobiologischen Funktionsprinzipen zur Gestaltung und zum Aufbau des visuellen Objekterkennungssystems bleiben jedoch viele Fragen zum optimalen Entwurf des konkreten Systems offen.

Das Entwurfsprinzip der Natur für die Grundstruktur des menschlichen visuellen Wahrnehmungssystems ist die Evolution. Innerhalb der phylogenetischen Entwicklung des Menschen wurde beginnend von den ersten Einzellern über frühe Hominiden bis hin zum Menschen der visuelle Kortex – Sitz der optischen Wahrnehmung – aufgebaut. Eng verbunden mit der Phylogenese ist die Ontogenese, der Entwicklungsprozess einer einzelnen befruchteten Eizelle zum ausgewachsenen Lebewesen. Dieser Vorgang ist weit mehr als das bloße Abarbeiten eines genetischen Bauplans. Vielmehr ist die Ontogenese ein aktiver

Strukturierungs- und Lernprozess, der in Rückkopplung mit sensorischen Reizen verläuft [47, 28]. Insbesondere im visuellen Kortex spielt die Lernfähigkeit ein wichtige Rolle [31].

Die technische Umsetzung der Evolution in Form der Evolutionären Algorithmen hat demonstriert, dass sie erfolgreich in der Lage ist, künstliche neuronale Netze optimal zu entwerfen [52]. Allerdings ist die Zahl der Arbeiten, die evolutionäre Methoden zum Entwurf von biologisch inspirierten visuellen Erkennungssystemen nutzen, eher gering [16, 45, 44, 27]. Auch werden bei den Evolutionären Algorithmen vorwiegend Prinzipien der Phylogenese und nicht der Ontogenese simuliert.

Zusammenfassend kann festgestellt werden, dass der Entwurf vieler biologisch inspirierter visueller Erkennungsstrukturen nicht in einer integrierten Form vorliegt und oftmals noch einer manuellen Einstellung vieler freier Systemparameter bedarf. Darüber hinaus gehen die Anwendungen in der Regel nicht über eine Handschrifterkennung oder die Erkennung einfacher Zeichen hinaus. Um jedoch eine anspruchsvolle dreidimensionale Objekterkennung zu realisieren, ist die Systemkomplexität an eine Grenze gestoßen, die wirkungsvoll nur noch durch einen systematischen Entwurfsprozess zu handhaben ist. Dieser wird erschwert durch die – bei komplizierten Erkennungssystemen – sehr große Anzahl von benötigten Neuronen und die vorhandenen Systemnichtlinearitäten, die eine starke Verkoppelung paralleler Merkmalsstränge bewirken.

Die vorliegende Arbeit liegt an der Nahtstelle der angesprochenen drei Forschungsgebiete: der computergestützten Bildverarbeitung, der neurobiologischen Modellierung und der evolutionären Optimierung. Diese ergänzen und befruchten sich in einer Weise, die es erlaubt, die im folgenden Abschnitt gestellten Aufgaben wirkungsvoll zu lösen.

Ansatz und Ziele

Ziel dieser Arbeit ist es, einen Entwurfsprozess zu entwickeln, zu implementieren und zu analysieren, der den systematischen und optimalen Aufbau des zuvor genannten visuellen Objekterkennungssystems erlaubt. Das optimierte visuelle System soll schnell und robust Objekte aus unterschiedlichen Bilddomänen erkennen können. Die optimierten visuellen Systeme sollen in ihrem Aufbau untersucht und in Bezug auf ihre Generalisierungsfähigkeit evaluiert werden.

Das verwendete visuelle Verarbeitungssystem bringt gezielt Vorwissen aus dem Bereich der Neurobiologie ein. Durch die Integration von leistungsfähigen Verarbeitungsprinzipien, die durch das menschliche Sehsystem inspiriert sind, wird der große Gestaltungsraum aller möglichen Systementwürfe sinnvoll eingegrenzt. Die Festlegung der verbleibenden freien Parameter und Strukturelemente soll in dieser Arbeit mit Hilfe Evolutionärer Algorithmen erfolgen. Aus dieser Klasse von Methoden werden speziell die Evolutionsstrategien [29]

verwendet, welche sich durch eine Selbstadaption der mutativen Schrittweite auszeichnen.

Die gängigen Evolutionären Algorithmen simulieren im Wesentlichen die Phylogenese, also die Weiterentwicklung von Individuen von Generation zu Generation. In dieser Arbeit sollen hingegen auch Elemente der Ontogenese erstmals bei der evolutionären Optimierung eines visuellen Erkenners, der ein anspruchsvolles Erkennungsproblem mit realen dreidimensionalen Objekten zu lösen vermag, verwendet werden. Hierzu sollen unterschiedliche unüberwachte Lernverfahren zum Einsatz kommen, die in den Prozess der evolutionären Optimierung eingebettet sind. Ein biologisches Prinzip, das dabei angewendet werden soll, ist das der spärlichen Kodierung. Die verwendete Kopplung von evolutionärer Optimierung und lokalen Lernverfahren wird in dieser Arbeit als *indirekte* Kodierung bezeichnet.

Von besonderer Bedeutung bei der evolutionären Optimierung von neuronalen Strukturen zur Objekterkennung ist die Frage nach ihrer Generalisierungsfähigkeit. Im Allgemeinen versteht man darunter bei technischen Objekterkennungssystemen die Fähigkeit des Systems, nach dem Erlernen von Trainingsansichten auf davon unterschiedliche Testansichten eines Objektes generalisieren zu können. Ein optimaler Entwurf des Erkennungssystems soll diese Fähigkeit verbessern. Überprüft wird diese Fähigkeit meist auf problemspezifischen Objektdaten. Wünschenswert ist jedoch ein System, das nicht nur auf einer beim Entwurf verwendeten Datenbank eine gute Generalisierung aufweist, sondern diese Fähigkeit domänenübergreifend, d.h. auch auf unterschiedlichen Objektklassen, aufweist. Diese Form der Generalisierung wird in dieser Arbeit mit Generalisierung 2. Ordnung bezeichnet. Sie dient als ein weiteres Evaluierungskriterium für das evolutionär strukturierte visuelle System.

Diese Generalisierungseigenschaft soll in Verbindung mit der Forderung nach Robustheit in die evolutionäre Optimierung eingebracht werden. Zwei Methoden werden hierzu untersucht: Zum einen soll eine Verallgemeinerung des Fitnessmaßes neben der reinen Erkennungsleistung die Konfidenz der Klassifikationsentscheidung mit aufnehmen, zum anderen wird durch die Simulation einer veränderlichen Bildumgebung die Entwicklung eines visuellen Systems mit allgemeiner einsetzbaren Merkmalen unterstützt.

Bei der Anwendung von stochastischen Optimierungsalgorithmen wie den Evolutionsstrategien kann das Erreichen des globalen Optimums nicht garantiert werden. Wird die Aufgabe nicht vollständig gelöst, so kann das an der Lösungsstruktur (hier dem visuellen System) oder aber an der Konvergenz des Algorithmus in lokalen Optima liegen. Um die Leistungsfähigkeit des visuellen Systems im Zusammenspiel mit der vorgeschlagenen evolutionären Optimierung genauer untersuchen zu können, soll ein generatives Modell zur Erzeugung von hierarchischen Musterdatenbanken entwickelt werden. Durch die damit ermöglichte Erzeugung von Mustern können gezielt Erkennungsaufgaben konzipiert werden. Der entscheidende Vorteil hierbei ist, dass von vornherein

die prinzipielle Lösbarkeit der gestellten Aufgabe mit der zu optimierenden visuellen Struktur bekannt ist.

Zum besseren Verständnis der Wirkungsweise sowohl der direkt als auch der indirekt kodierten evolutionären Optimierung (mittels der Einbettung unüberwachter Lernverfahren in Anlehnung an die Ontogenese) werden die unterschiedlichen Entwurfsverfahren und die jeweils erzeugten visuellen Systeme vergleichend analysiert. Zur weiteren Einordnung der Güte der optimierten visuellen Systeme sollen die erzielten Erkennungsergebnisse denen anderer leistungsfähiger Erkennungssysteme gegenübergestellt werden.

Aufbau der Arbeit

In Kapitel 2 wird kurz auf die in dieser Arbeit verwendeten Grundlagen eingegangen. Dazu wird zunächst die biologisch motivierte visuelle Merkmalshierarchie erläutert. Auf dieser basiert das visuelle System zur Objekterkennung, das im Anschluss daran vorgestellt wird. Leistungskritische Bestandteile des visuellen Systems werden als Ziele eines optimalen Entwurfsprozesses identifiziert. Nach einem kurzen Überblick über den Stand der Forschung bezogen auf die Optimierung von biologisch motivierten Erkennungssystemen wird das im nachfolgenden eingesetzte Entwurfsverfahren erklärt, das auf Evolutionsstrategien basiert. Danach wird auf eine Kopplung von evolutionären Entwurfsverfahren und lokalem Lernen zusammen mit einem Überblick über vorliegende Arbeiten eingegangen.

In Kapitel 3 wird die evolutionäre Optimierung des visuellen Systems dargestellt. Als Ausgangspunkt wird die Leistungsfähigkeit von Standardverfahren auf der exemplarisch untersuchten Objekterkennungsaufgabe ermittelt. Als erste Stufe der evolutionären Optimierung des Systems wird die Kodierung der zu bestimmenden Systemparameter und Strukturen dargelegt. Anschließend werden der Aufbau des evolutionären Entwurfsverfahrens erklärt und die Ergebnisse der Optimierung erläutert. In einem weiteren Schritt wird das in dieser Arbeit in besonderem Maße untersuchte Konzept der Generalisierung 1. und 2. Ordnung vorgestellt. Die Erkennungsleistung der optimierten visuellen Systeme wird im Hinblick darauf erneut analysiert. Ebenfalls untersucht werden die von der Optimierung gefundenen Systemparameter und Kombinationsmerkmale. Zum Vergleich der optimierten Kombinationsmerkmalsbänke wird ein Abstandsmaß definiert und zur Analyse eingesetzt. Im letzten Teil dieses Kapitels werden zwei unterschiedliche Verfahren zur Verbesserung der Generalisierung 2. Ordnung vorgestellt. Die erste Methode basiert auf einer Verallgemeinerung des Fitnessmaßes zu einem kontinuierlichen Evaluationsmaß, das neben der reinen Erkennungsleistung die Konfidenz der Entscheidung mit einbezieht. Die zweite Methode zur Erhöhung der Robustheit beruht auf einer gezielten Variation der visuellen Aufgabenstellung. Durch diese werden visuelle Systeme bevorzugt, die tendenziell eine größere Allgemeinheit in Bezug

auf die Ausbildung von visuellen Merkmalen aufweisen.

In Kapitel 4 wird die Leistungsfähigkeit des visuellen Systems im Zusammenspiel mit dem evolutionären Entwurfsverfahren anhand einer künstlich erzeugten Musterdatenbank untersucht. Die mit Hilfe des sogenannten generativen Modells erstellten hierarchischen Muster bilden die Basis einer wohldefinierten Erkennungsaufgabe, für die ein optimales visuelles System bekannt ist. Nach der Erzeugung einer beispielhaften Musterdatenbank wird die evolutionäre Optimierung dazu verwendet, das hierarchische visuelle System (das durch die Ähnlichkeit des Aufbaus mit dem generativen System dazu prädestiniert ist diese Aufgabe zu lösen) zur Lösung der Aufgabe anzupassen. Vergleichend dazu wird ein Multi-Layer-Perceptron zur Lösung derselben Aufgabe trainiert.

In Kapitel 5 wird ein neuartiges Verfahren, das auf einer Kopplung von Evolution und lokalem unüberwachtem Lernen basiert, zum Entwurf des visuellen Systems eingeführt. Die Einbettung des Lernens in den Prozess der evolutionären Anpassung geschieht in Anlehnung an die Einbettung der Ontogenese innerhalb der Phylogenese. Die unterschiedlichen zur Verwendung kommenden Lernverfahren werden dargestellt und in Bezug auf ihre Leistungsfähigkeit innerhalb der evolutionären Optimierung untersucht. Die erzielten Ergebnisse werden diskutiert und die gefundenen visuellen Systeme analysiert. In einem nächsten Schritt wird das neuartige indirekt kodierte evolutionäre Entwurfsverfahren mit dem in Kapitel 3 vorgestellten direkt kodierten evolutionären Verfahren verglichen. Die Erkennungsergebnisse des optimierten visuellen Systems werden außerdem mit anderen gegenwärtigen Erkennungssystemen verglichen.

Im abschließenden Kapitel 6 werden die in der Arbeit gefundenen Ergebnisse kurz zusammengefasst und darauf aufbauend mögliche und Erfolg versprechende Richtungen zukünftiger Forschungen skizziert.

Kapitel 2

Grundlagen

In diesem Kapitel wird zunächst das in dieser Arbeit eingesetzte biologisch motivierte visuelle System vorgestellt. Nach der Beschreibung des Systems wird die damit durchgeführte Objekterkennung betrachtet. In dem zweiten Teil dieses Grundlagenkapitels wird auf Methoden des evolutionären Entwurfs von allgemeinen biologisch motivierten optischen Objekterkennern eingegangen. Dazu werden auch andere Entwurfs- bzw. Optimierungsmethoden von solchen Systemen kurz beschrieben. Anschließend werden die Grundlagen für die in dieser Arbeit verfolgte Methode des Systementwurfs mit Evolutionsstrategien bereitgestellt. In dem dritten Teil dieses Kapitels wird der Stand der Forschung bezüglich der Kopplung von evolutionären Methoden und lokalem Lernen kurz skizziert.

2.1 Biologisch motiviertes visuelles System

Zur visuellen Objekterkennung wird in dieser Arbeit ein von biologischen Grundprinzipien motiviertes System verwendet. Dieses von Wersing und Körner [49, 50] entwickelte System gehört zu der Gruppe der ansichtsbasierten Erkennungssysteme. Es basiert im Wesentlichen darauf, dass von jedem zu erkennenden Objekt unterschiedliche prototypische Ansichten in einen hochdimensionalen Merkmalsraum transformiert werden. Um anschließend unbekannte Ansichten eines so gelernten Objektes wiedererkennen zu können, wird die zu identifizierende Ansicht in denselben Merkmalsraum transformiert. Die nachfolgende Klassifikationsaufgabe findet jetzt in diesem Raum mittels einfacher Standardklassifikatoren statt. In dieser Arbeit wird hierfür im Allgemeinen ein Nächster-Nachbar-Klassifikator verwendet. Im Folgenden wird nun erklärt, auf welche Weise und nach welchen biologisch motivierten Grundprinzipien das visuelle System die Transformation der Eingangsbilder in den Merkmalsraum vornimmt.

Wie der menschliche visuelle Kortex ist das visuelle System hierarchisch aufgebaut und lokal vernetzt. Mehrere Schichten von künstlichen Neuronen

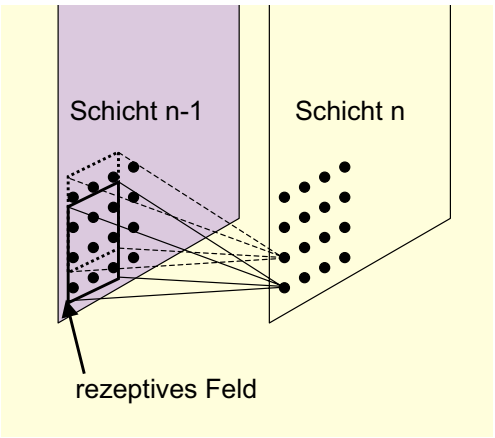

Abbildung 2.1: Schematische Darstellung der Architektur der Verschaltung von zwei neuronalen Schichten über rezeptive Felder.

folgen sukzessive aufeinander. Die einzelnen Neuronen sind dabei über sogenannte *rezeptive Felder* (RF) miteinander verschaltet. Die visuellen Informationen verlaufen dabei stets von der ersten Schicht über alle Zwischenschichten zur letzten Schicht. Schematisch ist diese Architektur in Abbildung 2.1 dargestellt. Die rezeptiven Felder von benachbarten Neuronen können sich dabei auch überlappen.

Im visuellen Kortex wurden sogenannte *Simple-* und *Complex-Cells* gefunden. Während erstere besonders stark auf ein spezifisches optisches Merkmal an einer bestimmten Stelle reagieren, sind letztere weitaus weniger sensitiv bezüglich der Lage eines Merkmals [18]. Auch in dem vorgestellten künstlichen visuellen System sind Neuronen implementiert, die auf bestimmte Merkmale an bestimmten Orten im visuellen Feld sensitiv sind, und andere, die auf die gleichen Merkmale in einem größeren Bereich (rezeptiven Feld) sensitiv sind. Durch letztere wird ein bestimmtes Maß an räumlicher Invarianz gegenüber einer Translation von optischen Merkmalen erreicht. Beide Neuronenarten wechseln sich in der Verarbeitung sukzessive ab. So folgt auf eine sogenannte *S-Schicht* (für Simple-Cells) eine sogenannte *C-Schicht* (für Complex-Cells).

Die ersten optischen Merkmale, auf die der visuelle Input untersucht wird, sind sogenannte *Gabor-Merkmale*. Diese konnten auch im biologischen Sehsystem als frühe Merkmale nachgewiesen werden [48]. Weiter konnte gezeigt werden, dass höhere visuelle Kortexregionen aufbauend auf die ersten einfachen Kantenmerkmale sensitiv auf immer komplexere Merkmale sind [14]. Wie der visuelle Kortex letztlich die Erkennung von Objekten exakt bewerkstelligt, ist nicht bekannt, jedoch wird angenommen, dass hierarchisch aufgebaute komplexe Merkmale hierbei eine wichtige Rolle spielen.

2.1.1 Visuelle Merkmalshierarchie

Im Zentrum des von Wersing und Körner [49, 50] vorgestellten visuellen Erkennungssystems steht eine biologisch inspirierte visuelle Merkmalshierarchie. Diese ist vergleichbar mit dem von Fukushima entwickelten *Neocognitron* [8]. Sie basiert im Wesentlichen auf der sequentiellen Hintereinanderschaltung von zwei unterschiedlichen neuronalen Verarbeitungsschichten: Eine Schicht von Neuronen, die sensitiv auf elementare räumliche Merkmale reagiert, wird gefolgt von einer Schicht von „Pooling"-Neuronen. Diese sorgen für eine Erhöhung der Translationsinvarianz, indem sie in einem gewissen räumlichen Bereich unabhängig von der Lage der Merkmale aktiviert werden können. Durch eine Abfolge dieser Schichten kommt es dazu, dass die Neuronen aus höheren Schichten mittelbar über immer größere rezeptive Felder verfügen, d.h. mit immer größeren räumlichen Bildbereichen vernetzt sind. Zusätzlich verfügt die visuelle Merkmalshierarchie über eine Reihe von nichtlinearen Verarbeitungsschritten. So kontrolliert beispielsweise eine *Winner-Take-Most* Charakteristik die laterale Inhibition der Aktivierungen paralleler Merkmalsebenen innerhalb einer Schicht. Dies unterscheidet sich von der Maximums-Nichtlinearität, die von Riesenhuber und Poggio als Erweiterung des Neocognitrons vorgeschlagen wurde [30]. Zur Bestimmung von höheren Zwischenschichtmerkmalen, den *Kombinationsmerkmalen*, wird ein Verfahren zur spärlichen Kodierung vorgeschlagen. Die Leistungsfähigkeit des visuellen Systems im Vergleich zu anderen gegenwärtigen Erkennern wird mit Hilfe von Objekterkennungsaufgaben demonstriert. Zu erkennen sind zum einen reale dreidimensionale Objekte und zum anderen Bilder von unterschiedlichen menschlichen Gesichtern [50].

Eine schematische Darstellung der visuellen Hierarchie ist in Abbildung 2.2 zu finden. Die unterste Schicht ist der visuelle Input selbst. Das Inputbild sei gegeben durch den Bildvektor $\mathbf{I}$. Es folgen die Schichten S1, C1, S2 und C2, die sich weiter in sogenannte *Ebenen* unterteilen lassen. Basierend auf dem Inputbildvektor werden die sogenannten *Aktivierungen* aller folgenden Schichten berechnet. Die Aktivierung der Schicht S1 wird mit Schichtaktivierung $\bar{\mathbf{s}}_1 = (\mathbf{s}_1^1, \ldots, \mathbf{s}_1^{P_1})$ bezeichnet. Hierbei ist P_1 gleich der Anzahl der Ebenen der ersten Schicht. Weiter wird der Wert eines einzelnen Neurons an der Position (x, y) in Ebene l und Schicht i mit $s_i^l(x, y)$ bezeichnet. Der Schicht S1 sind P_1 lokale Filter $\mathbf{w}_1^l$ zugeordnet, die alle parallel von den jeweiligen Bereichen des Inputbildes $\mathbf{I}$ aktiviert werden. Diese Filter werden auch als *Merkmalsfilter*, *Merkmale* oder *Features* bezeichnet. Die Inputschicht wird mit jedem dieser Filter gefaltet:

$$q_1^l(x, y) = |\mathbf{w}_1^l(x, y) * \mathbf{I}|. \tag{2.1}$$

Hierbei bezeichnet * das Skalarprodukt und $\mathbf{w}_1^l(x, y)$ ist der Rezeptive-Feld-Vektor des Merkmals l. Dieser beschreibt die „Einbettung" von $\mathbf{w}_1^l$ in einen Nullvektor mit der gleichen Länge wie $\mathbf{I}$ entsprechend dem rezeptiven Feld des Merkmals und der zu berechnenden Position (x, y). Das bedeutet für ein 10×10

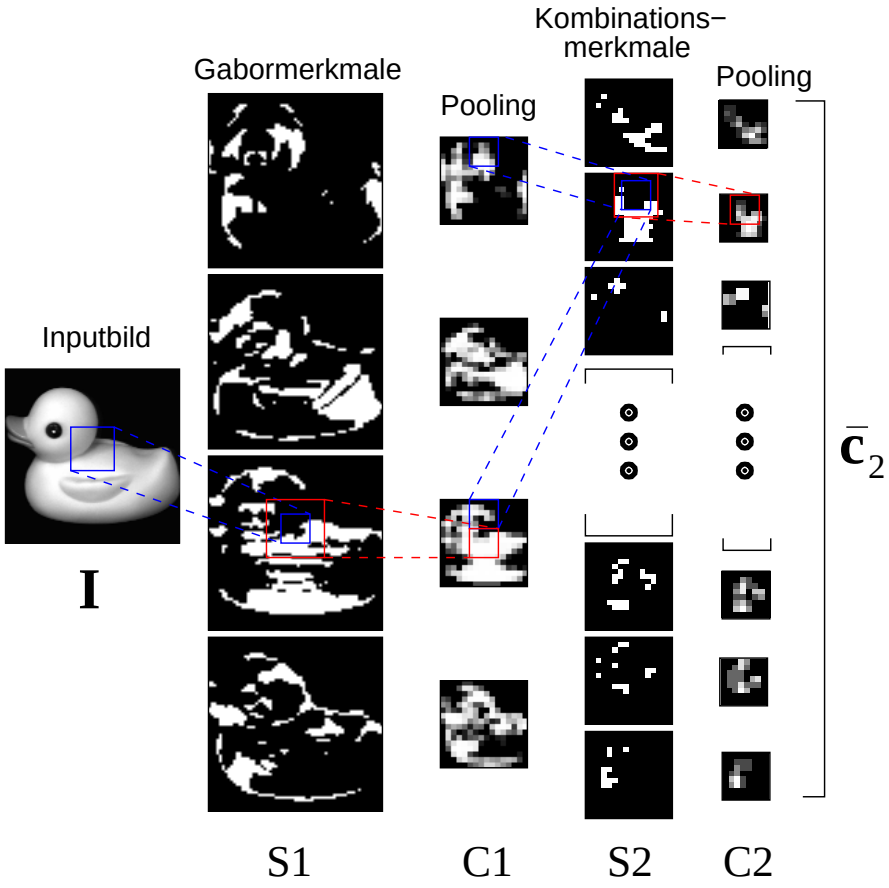

Abbildung 2.2: Schematische Darstellung der visuellen Hierarchie. Das Inputbild **I** ist ein 64×64 Pixel Bild. Die S1-Schicht besteht aus vier Ebenen, denen unterschiedlich orientierte Gabormerkmale zugeordnet sind. Die Dimension jeder Ebene ist 64×64. Die C1-Schicht entsteht durch Pooling und Unterabtastung der S1-Schicht (Dimension einer Ebene: 16×16). Die S2-Schicht besteht aus L Ebenen, denen jeweils ein *Kombinationsmerkmal* (KM) zugeordnet ist. Dadurch hat jedes S2-Neuron potentiell Verbindungen zu C1-Neuronen in allen Ebenen aus einem räumlichen Bereich. Die C2-Schicht geht hervor aus einem Pooling und einer Unterabtastung der S2-Schicht auf eine Dimension von 8×8. Auf diese Weise wird ein Inputbild **I** in einen Schichtaktivierungsvektor $\bar{\mathbf{c}}_2$ transformiert.

Pixel großes Inputbild $\mathbf{I}$ einen 100-dimensionalen Vektor $\mathbf{w}_1^l(x,y)$. Diese Art der Architektur gewährleistet, dass jedes Neuron der Schicht S1 nur auf das rezeptive Feld im Inputbild „sieht". Alle Neuronen der Merkmalsebene l haben die gleiche Struktur des rezeptiven Feldes und auch dieselben Gewichte $\mathbf{w}_1^l$. Auf diese Weise werden die Gewichte eines Merkmals für alle Neuronen einer Ebene verwandt und man erhält eine sogenannte *Weight-Sharing* Architektur. Nach dieser Faltungsoperation wird ein sogenannter *Winner-Take-Most* (WTM) Schritt durchgeführt. Hierbei werden alle Aktivierungen gemäß einer Kompetition unter räumlich gleichen Neuronen unterschiedlicher Ebenen gewichtet. Der größte Wert an einer räumlichen Position bleibt erhalten, wohingegen kleinere Werte anderer Ebenen linear abgeschwächt werden:

$$r_1^l(x,y) = \begin{cases} 0 & \text{wenn } \frac{q_1^l(x,y)}{M} < \gamma_1 \text{ oder } M = 0, \\ \frac{q_1^l(x,y) - M\gamma_1}{1-\gamma_1} & \text{sonst.} \end{cases} \tag{2.2}$$

Hierbei entspricht M dem Maximalwert an der Stelle (x,y) über alle P_1 Ebenen der ersten Schicht: $M = \max_l q_1^l(x,y)$. Das Ergebnis des WTM-Schrittes wird mit $r_1^l(x,y)$ bezeichnet. Mittels des Parameters γ_1 wird die Stärke der Kompetition innerhalb der Schicht S1 gesteuert. γ_1 ist hierbei auf das halboffene Intervall $[0,1[$ beschränkt. Für Werte nahe 0 ist die WTM-Operation beinahe ausgeschaltet und alle Werte werden praktisch unverändert weiterpropagiert. Für Werte nahe 1 hingegen werden alle Werte außer dem größten Wert[1] an einer räumlichen Position auf 0 reduziert. Durch diesen Mechanismus wird das biologische Prinzip der latenzbasierten Kompetition in das Modell eingebracht [50].

Nach dem WTM-Schritt werden alle Aktivierungen noch einer Schwellwertoperation mit einem globalen Schwellwert θ_1 unterzogen:

$$s_1^l(x,y) = \mathrm{H}\big(r_1^l(x,y) - \theta_1\big). \tag{2.3}$$

H bezeichnet die Heavysidefunktion mit $\mathrm{H}(x) = 1$ für $x \geq 0$ und $\mathrm{H}(x) = 0$ für $x < 0$. Als Merkmale der S1-Schicht werden vier sogenannte *Gaborfilter* verwendet. Diese haben die Aufgabe, unterschiedlich orientierte Kanten bzw. Helligkeitsübergänge im Eingangsbild zu detektieren. Gegeben sind die Filterwerte durch:

$$\begin{aligned} w(x,y) = \exp\left[-\tfrac{1}{2}f^2\left(\left(\tfrac{x\cos(\alpha)+y\sin(\alpha)}{r}\right)^2 + \left(\tfrac{-x\sin(\alpha)+y\cos(\alpha)}{r}\right)^2\right)\right] \\ \cdot \sin\left(f(x\cos(\alpha) + y\sin(\alpha))\right), \end{aligned} \tag{2.4}$$

mit $f = 12, r = 1.5$ und $\alpha = 0, \pi/4, \pi/2, 3\pi/4$. Auf diese Weise ist die erste Ebene sensitiv auf vertikal orientierte Kanten und die drei folgenden Ebenen auf Orientierungen, die um jeweils 45° im mathematisch positiven Sinne weiter gedreht sind.

[1] Dies können auch mehrere gleich große Werte sein.

Die Aktivierungen der C1-Schicht sind gegeben durch:

$$c_1^l(x,y) = \tanh\left(\mathbf{g}_{\text{Gauß},1}(x,y) * \mathbf{s}_1^l\right), \tag{2.5}$$

wobei $\mathbf{g}_{\text{Gauß},1}(x,y)$ eine normalisierte Gaußfunktion mit einer Standardabweichung von σ_1 ist. Mit *tanh* ist der Tangens hyperbolicus bezeichnet. Weiter ist zu bemerken, dass die Faltung mit dem Gaußkern nur mit räumlich jedem vierten Neuron der vorangehenden Schicht durchgeführt wird. Durch diese Unterabtastung haben die Ebenen der C1-Schicht eine Größe von nur noch 16×16 Neuronen. Durch die Faltung mit dem Gaußkern wird eine räumliche Verwischung von Aktivierungen erreicht. Diese Operation wird auch als *Pooling* bezeichnet. Es bewirkt eine gewisse Insensitivität gegenüber kleinen Translationen von Bildkanten, die um so größer wird je größer σ_1 eingestellt ist. Das Pooling wird für alle Ebenen gleich stark durchgeführt.

Die Merkmale der Zwischenschicht S2 reagieren sensitiv auf räumliche Kombinationen der Merkmale der vorhergehenden C1-Schicht. Aus diesem Grunde spricht man hier auch von *Kombinationsmerkmalen* oder *Combination-Features*. Durch die räumliche Kombination von unterschiedlich orientierten Kantenmerkmalen reagieren die Merkmale in der S2-Schicht zum Beispiel auf Ecken und T-Verbindungen sensitiv. Die Wahl von geeigneten Kombinationsmerkmalen ist von entscheidender Bedeutung für die Funktionsweise des visuellen Objekterkennungssystems. Die Gewichte eines Merkmals der Schicht S2 sind in dem Schicht-Gewichtsvektor $\bar{\mathbf{w}}_2^l = (\mathbf{w}_2^{l1}, \ldots, \mathbf{w}_2^{lP_1})$ zusammengefasst. Hierbei ist $P_1 = 4$ die Anzahl der Ebenen der vorangehenden Schicht C1. $\mathbf{w}_2^{lk}(x,y)$ ist der Rezeptive-Feld-Vektor eines S2-Neurons des Merkmals l an der Position (x,y). Er beschreibt die Verbindungen dieses Neurons mit den C1-Neuronen der Ebene k. Das Ergebnis der Faltung eines Schicht-Gewichtsvektors mit allen Ebenen der C1-Schicht ist gegeben durch:

$$q_2^l(x,y) = \bar{\mathbf{w}}_2^l(x,y) * \bar{\mathbf{c}}_1. \tag{2.6}$$

Nach der Berechnung aller $q_2^l(x,y)$-Werte werden diese wie die $q_1^l(x,y)$-Werte in Gleichung (2.2) einer WTM-Operation unterzogen:

$$r_2^l(x,y) = \begin{cases} 0 & \text{wenn } \frac{q_2^l(x,y)}{M} < \gamma_2 \text{ oder } M = 0, \\ \frac{q_2^l(x,y) - M\gamma_2}{1-\gamma_2} & \text{sonst.} \end{cases} \tag{2.7}$$

Die Kompetition in dieser Schicht wird durch den Parameter $\gamma_2 \in [0,1[$ gesteuert. Die finale Aktivierung der S2-Schicht ist das Ergebnis einer Schwellwertoperation (analog zu Gleichung (2.3)):

$$s_2^l(x,y) = \mathrm{H}\big(r_2^l(x,y) - \theta_2\big), \tag{2.8}$$

mit dem globalen Schwellwertparameter θ_2. Analog zu der Berechnung der C1-Aktivierungen aus den S1-Aktivierungen in Gleichung (2.5) werden jetzt

die C2-Aktivierungen aus den S2-Aktivierungen berechnet:

$$c_2^l(x,y) = \tanh\left(\mathbf{g}_{\text{Gauß},2}(x,y) * \mathbf{s}_2^l\right). \tag{2.9}$$

Hierbei hat der Gaußkern die Standardabweichung σ_2. Wie auch bei der C1-Schicht wird zusammen mit Gleichung (2.9) eine Unterabtastung realisiert. Im Falle der C2-Schicht jedoch nur um den Faktor 2, d.h., dass die Dimension der Ebenen 8×8 Neuronen beträgt. Die zweimal durchgeführte Unterabtastung und damit einhergehende Dimensionsreduktion der Schichtebenen ist wichtig, um dem gleichzeitigen Anstieg der Anzahl der Ebenen mit jedem verwendeten Merkmal entgegenzuwirken. Mit der Aktivierung der C2-Schicht ist die letzte Schicht der visuellen Verarbeitungshierarchie erreicht.

Gegenstand des Entwurfs

Ziel der vorliegenden Arbeit ist es, ein Entwurfsverfahren zu entwickeln, das in der Lage ist, ein biologisch motiviertes visuelles Objekterkennungssystem zu optimieren. Bei der Optimierung steht neben einer kurzen Erkennungszeit vor allem eine geringe Fehlklassifikationsrate im Vordergrund. Der Entwurf soll nach Möglichkeit so robust sein, dass das visuelle System diese Leistung unabhängig von der verwendeten Bilddatenbank erbringen kann. Exemplarisch wird hierzu die Klassifikationsrate des vorgestellten visuellen Systems auf einer Bilddatenbank bestehend aus realen dreidimensionalen Objekten optimiert. Bei dem optimalen Entwurf des Systems sollen die beschriebenen biologisch motivierten Grundverarbeitungsprinzipen erhalten bleiben. Gegenstand des Entwurfs ist der verbleibende Gestaltungsraum. Innerhalb dieses Raumes ist immer noch eine sehr große Anzahl von unterschiedlichen Systemen möglich. Dieser Gestaltungsraum wird durch viele unterschiedliche Systemparameter und -strukturen aufgespannt. Um den Entwurfsprozess möglichst effizient durchführen zu können, ist es entscheidend, geeignete Systembestandteile zu definieren, innerhalb derer die Suche stattfinden soll. Kennzeichen dieser Strukturen und Parameter ist, dass ihre Auslegung besonders kritisch in Bezug auf die Leistungsfähigkeit des Systems ist. Was die Bestimmung der Strukturen und Parameter zumeist erschwert, ist ihre starke gegenseitige Verkopplung, welche eine getrennte Suche verhindert. Im Folgenden wird erläutert, welche Bestandteile des visuellen Systems – in dieser Arbeit – in den Entwurfsprozess eingehen sollen.

Systemnichtlinearitäten

Maßgebliche Parameter für die Funktionsweise des visuellen Systems sind die drei *Systemnichtlinearitäten*: laterale Kompetition (vgl. Gleichungen (2.2) und (2.7)), Schwellwertbildung (vgl. Gleichungen (2.3) und (2.8)) und Pooling (vgl. Gleichungen (2.5) und (2.9)). Da die entsprechenden Operationen in aufeinander folgenden Schichten ausgeführt werden und nachfolgende Operationen auf

den Ergebnissen vorangehender aufbauen, beeinflussen sich diese Nichtlinearitäten gegenseitig. Die Funktionsweise dieser Nichtlinearitäten ist im visuellen System durch die folgenden Parameter gesteuert. So ist die laterale Kompetition durch die WTM-Parameter γ_1, γ_2, die Schwellwertoperationen durch die Parameter θ_1, θ_2 und das Pooling durch die Standardabweichungen der Gaußkerne σ_1, σ_2 bestimmt. Die beiden Poolingweiten σ_1 und σ_2 sind überdies strukturbestimmend, da durch sie festgelegt wird, wie weit die lokale Vernetzung der rezeptiven Felder der C1- und C2-Schicht reicht. Es ist festzustellen, dass schon kleine Veränderungen der Parameterwerte der Nichtlinearitäten zu einer stark veränderten Funktionsweise des Systems führen können. Vorwissen, wie diese Werte einzustellen sind, existiert nicht. Aus diesen Gründen sind die Nichtlinearitäten – definiert durch die obigen Parameterwerte – geeignete Kandidaten für eine Auslegung mit Hilfe eines systematischen Entwurfsverfahrens.

Kombinationsmerkmale

Die optimale Auslegung und die Anzahl L der höheren Merkmale des visuellen Systems, die Kombinationsmerkmale, sind ebenfalls sowohl eine offene Frage als auch ein kritischer Punkt für die Funktionsweise des Systems. Ein wichtiges Konzept innerhalb eines hierarchisch organisierten visuellen Systems ist die Reduktion von Redundanz. So betont Barlow [2] die Wichtigkeit der neuronalen Verarbeitung zur Reduktion der statistischen Abhängigkeiten innerhalb der einzelnen Elemente des visuellen Verarbeitungsstromes. Diese wird zum einen durch die Einführung einer spärlichen und übervollständigen Repräsentation [26], zum anderen durch die Anwendung der Independent-Component-Analysis (ICA) [3] eingebracht. Hoyer und Hyvärinen [17] verwenden die sogenannte Methode des *non-negative-Sparse-Coding*, um höhere Merkmale zu bilden. Dieses Verfahren wird auch von Wersing und Körner innerhalb der vorgestellten Hierarchie zum Entwurf der Kombinationsmerkmale verwendet [49] und in [50] erweitert. Alle aufgeführten Verfahren wenden demnach unüberwachte Lernregeln auf eine Menge von visuellen Inputs an mit dem Ziel, charakteristische Strukturen aus den Eingangsdaten zu extrahieren. Um allerdings die Funktionsweise des gesamten visuellen Systems zu garantieren, kann der Entwurf der Kombinationsmerkmale nicht isoliert betrachtet werden. Vielmehr ist er in enger Abstimmung mit dem Entwurf des Gesamtsystems durchzuführen. Weiter kompliziert wird der Entwurf der Merkmale durch die große Zahl der Parameterwerte $\bar{\mathbf{w}}_2^l$, durch die sie beschrieben werden. Durch diese Art der Repräsentation ist die Dimensionalität des Suchraumes eines Entwurfsverfahrens entsprechend hoch, was im Allgemeinen auch die Suche erschwert.

Nicht optimiert werden sollen die Merkmale der ersten Schicht $\bar{\mathbf{w}}_1^l$, die Gabormerkmale (vgl. Gleichung (2.4)). Ihre spezielle Parametrisierung ist weitgehend festgelegt durch die räumliche Auflösung der zu erkennenden Objekte. Dadurch dass die Gabormerkmale nicht Bestandteil des Entwurfs sind, ergibt

sich auch der Vorteil, dass eine Evaluation des visuellen Systems mit einem geringeren Rechenaufwand durchgeführt werden kann. Das liegt daran, dass die rechenaufwendige Faltung jedes Eingangsbildes (vgl. Gleichung (2.1)) nicht bei der Bewertung jedes Systementwurfes durchzuführen ist. Das hat insbesondere einen Vorteil bei Entwurfsverfahren, die eine sehr große Zahl von Bewertungen von unterschiedlichen Entwürfen benötigen.

2.1.2 Objekterkennung mit dem visuellen System

Im Folgenden wird beschrieben, wie die im letzten Abschnitt betrachtete visuelle Merkmalshierarchie zur Erkennung von Objekten eingesetzt werden soll. Das damit entstehende Gesamterkennungssystem wird als *visuelles System* bezeichnet. Ziel ist es, das visuelle System so mit einigen wenigen Ansichten je eines Objektes zu trainieren, dass nie gesehene Ansichten dieses Objektes korrekt wiedererkannt werden. Die zum Belehren des Systems benutzten Ansichten werden im Folgenden als *Trainingsansichten* und die zum Testen der Erkennung benutzten Ansichten als *Testansichten* bezeichnet. Die Objekterkennung mit der oben beschriebenen visuellen Hierarchie funktioniert folgendermaßen: Der Lernschritt besteht darin, die C2-Aktivierungen der Trainingsansichten aller Objekte zu berechnen und zusammen mit der Labelinformation – um welches Objekt es sich handelt – abzuspeichern. Um eine Testansicht zu klassifizieren, d.h. mit der korrekten Objektbezeichnung zu assoziieren, wird dieses Bild ebenfalls in den C2-Raum abgebildet. In diesem Raum wird der am nächsten benachbarte Punkt (in Euklidischer Metrik) bestimmt, und dessen durch das Training vorhandene Labelinformation dem zu klassifizierenden Bild zugeordnet. Eine schematische Darstellung dieser Erkennungsmethode findet sich in Abbildung 2.3.

Bei dieser Form der Objekterkennung legt man also im Wesentlichen zugrunde, dass die visuelle Hierarchie in der Lage ist, die im Ansichtsraum vorliegenden Objekte in einen solchen Raum zu transformieren (den C2-Raum), in dem die Nachbarschaftsverhältnisse eine Klassifikation mit sehr einfachen Mitteln, wie z.B. der Nächsten-Nachbar(NN)-Suche, zulassen. Natürlich ist es auch möglich, basierend auf der C2-Repräsentation kompliziertere Klassifikatorstrukturen, wie z.B. Multi-Layer-Perceptrons (MLP), anzuwenden. Im Rahmen dieser Arbeit liegt jedoch der Fokus auf der evolutionären Optimierung der visuellen Hierarchie. Daher wird im Folgenden stets die NN-Suche zur Klassifikation benutzt. Ein wesentlicher Vorteil bei der Verwendung dieser sehr einfachen, aber leistungsstarken Methode ist die Tatsache, dass sie ohne zeitraubende Lernverfahren direkt eine Klassifikation durchführen kann. Diese Zeiteinsparung wird in der späteren evolutionären Optimierung von Bedeutung sein.

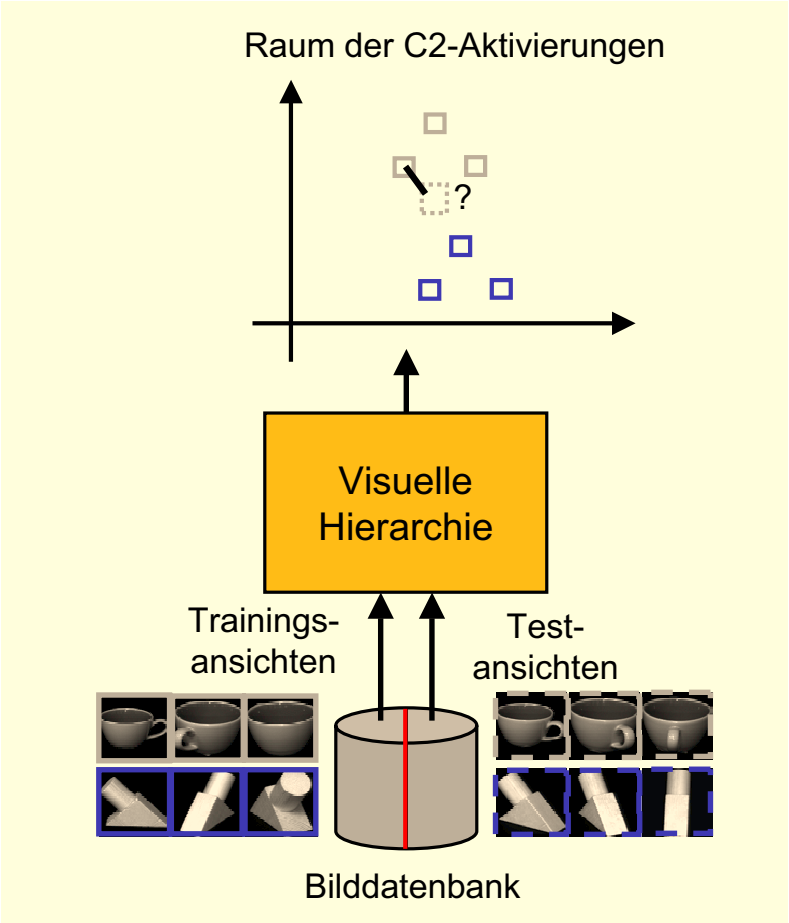

Abbildung 2.3: Schematische Darstellung der Objekterkennung mit Hilfe der visuellen Hierarchie. Diese transformiert die Bilddaten in den hochdimensionalen Raum der C2-Aktivierungen (dargestellt in zwei Dimensionen). Mit einer dort durchgeführten Nächsten-Nachbar-Suche können die Testansichten klassifiziert werden, indem ihnen der Objektlabel der nächsten Trainingsansicht zugeordnet wird.

Untersuchte Problemstellung

Zur Untersuchung der Lern-, der Leistungs- und der Generalisierungsfähigkeit des biologisch inspirierten visuellen Systems wird im Folgenden eine komplexe Erkennungsaufgabe von realen dreidimensionalen Objekten herangezogen. Von jedem Objekt liegt hierbei eine Anzahl von unterschiedlichen Bildansichten vor. Dem visuellen Klassifikationssystem wird eine geringe Anzahl von Bildansichten je Objekt zusammen mit der Objektbezeichnung, dem Label, zum Training übergeben. Die Klassifikationsaufgabe für das System besteht anschließend darin, andere, bisher nicht gesehene Bildansichten der Objekte mit den korrekten Labeln zu versehen und damit zu erkennen, bzw. zu klassifizieren. Nicht Teil der Aufgabe ist der Vorverarbeitungsschritt der Segmentierung der Objekte. Auch werden die Objekte zu einem gewissen Umfang als größennormiert angenommen. Eine Randbedingung an die Objekterkennung ist, dass Bilder nach sehr kurzer Verarbeitungszeit erkannt werden können. D.h., die Erkennungszeit soll unter einer Sekunde auf einem handelsüblichen Standardrechner sein (In dieser Arbeit wurde hierzu z.B. ein Pentium III mit 850 MHz verwendet.). Durch diese schnelle Erkennung wird die Einbindung des Systems in eine Reihe von anspruchsvollen Applikationen in den Bereichen Robotik, Mensch-Maschine-Interfaces und Fahrerassistenzsystemen ermöglicht.

Zur Weiterentwicklung und Optimierung des beschriebenen allgemeinen visuellen Erkennungssystems wird im Rahmen dieser Arbeit im Wesentlichen eine klar definierte visuelle Aufgabenstellung herangezogen. Die allgemeinen Methoden des Entwurfsprozesses sollen am Beispiel der Optimierung des visuellen Systems in Bezug auf diese Aufgabenstellung beispielhaft vorgeführt werden. Im folgenden Abschnitt wird diese visuelle Problemstellung genauer beschrieben. Sollten in später folgenden Kapiteln keine speziellen Angaben zu dem Aufbau einer untersuchten visuellen Problemstellung gemacht werden, so werden die im Folgenden beschriebenen Einstellungen verwendet.

In dieser Arbeit wird die Aufgabe der visuellen Objekterkennung am Beispiel der COIL Bilddatenbanken [25] untersucht. Die Datenbank COIL20 umfasst Bilder von 20 unterschiedlichen dreidimensionalen Objekten, wohingegen die COIL100 Datenbank 100 Objekte enthält. Ansonsten ist der Aufbau beider Bilddatenbanken analog. Je Objekt sind 72 Ansichten aus unterschiedlichen Blickwinkeln vorhanden. Die unterschiedlichen Ansichten wurden aufgenommen, während das Objekt um seine Längsachse um 360° gedreht wurde. Die 72 Ansichten sind in 5° Schritten aufgenommen. Die Aufnahme der Objekte erfolgte vor einem schwarzen Hintergrund, was einer Segmentierung nahe kommt. Die Bilder der COIL20 Datenbank liegen als Grauwertbilder vor. Die im Original farbig vorliegenden COIL100 Bilder wurden für die folgenden Untersuchungen in Graubilder konvertiert, wobei Weiß mit dem Wert 1.0 und Schwarz mit dem Wert 0.0 kodiert ist. Zusätzlich wurden alle Bilder von ursprünglich 128×128 Pixel auf 64×64 Pixel große Bilder herunterskaliert.

Abbildung 2.4: Objekte der COIL100 Datendank zusammen mit drei Trainingsansichten von zwei Objekten und drei der 24 Testansichten der entsprechenden Objekte.

In den folgenden Untersuchungen wurde mit drei Trainingsansichten und 24 Testansichten gearbeitet. Die Blickwinkel für die Trainingsansichten sind hierbei $\alpha_{\text{Train}} = 0°, 120°, 240°$ und die Winkel für die Testansichten sind $\alpha_{\text{Test}} = 5° + i \cdot 15°$ mit $i = 0, 1, 2, ..., 23$. Auf diese Weise wird gewährleistet, dass die Trainingsansichten von den Testansichten verschieden sind und dass die Ansichten beider Mengen gleichmäßig zwischen 0° und 360° verteilt liegen. Die Leistung, die jetzt von einem Objekterkenner zu bewältigen ist, ist die Generalisierung von den drei Trainingsansichten eines Objektes auf die 24 Testansichten. Abbildung 2.4 zeigt die 100 Objekte der COIL100 Datenbank zusammen mit drei Trainingsansichten von zwei Objekten und drei der 24 Testansichten der entsprechenden Objekte. Es ist zu bemerken, dass die Schwierigkeit der Erkennungsaufgabe sehr stark mit der Anzahl der zur Verfügung stehenden Trainingsansichten skaliert.

Prinzipiell ist davon auszugehen, dass für unterschiedliche Objekte eine unterschiedliche Anzahl von Trainingsansichten notwendig ist. So ist beispielsweise für ein Objekt, das annähernd rotationssymmetrisch bezüglich der Drehachse ist, unter Umständen ein einziges Trainingsbild ausreichend, um alle verbleibenden Testansichten korrekt klassifizieren zu können. Für ein anderes Objekt kann es sein, dass mindestens vier Ansichten dafür nötig sind. Hierbei ist auch vorstellbar, dass die Trainingsansichten nicht notwendigerweise gleichmäßig verteilt sein müssen, sondern dass bestimmte Ansichtsbereiche mit mehr Trainingsansichten abzudecken sind als andere. Für die weiteren Untersuchungen werden jedoch diese Aspekte nicht weiter betrachtet und stets drei Trainingsansichten unter den Ansichtswinkeln $\alpha_{\text{Train}} = 0°, 120°, 240°$ verwendet. Hierbei ist die fixe Anzahl von drei Ansichten als die absolute Untergrenze und damit als ein sehr hoher Schwierigkeitsgrad anzusehen, da sich manche Objekte in ihren unterschiedlichen Ansichten stark voneinander unterscheiden.

Es ist zu bemerken, dass von den 100 unterschiedlichen Objekten der COIL-100 Datenbank 17 Objekte bereits in der COIL20 Datenbank vorkommen. Aus diesem Grund definieren wir die COILselect Objektdatenbank, die nur aus den verbleibenden 83 Objekten der COIL100 besteht, die nicht in der COIL20 vorkommen. Auf diese Weise ist eine strikte Trennung der beiden Datenbanken COIL20 und COILselect gewährleistet, die in den weiteren Untersuchungen noch von Bedeutung sein wird. Das visuelle Erkennungsproblem, welches nun im Folgenden gelöst werden soll, ist die Erkennung bzw. Klassifikation der 24 Testbilder der COILselect Datenbank nach dem Training mit den drei Testbildern derselben Datenbank.

2.2 Evolutionäre Entwurfsverfahren

Das biologisch motivierte visuelle Erkennungssystem (vgl. Abschnitt 2.1) beruht in wichtigen Prinzipien auf Erkenntnissen der Neurobiologie [49, 50]. Zum

Entwurf des letztendlich technischen Erkenners sind jedoch noch die Parameter der Nichtlinearitäten und die Kombinationsmerkmale zu bestimmen. Beide Systembestandteile sind in enger gegenseitiger Abstimmung zueinander zu bestimmen, um eine optimale Funktionsweise des visuellen Erkennungssystems zu gewährleisten. Der Entwurf wird aus den folgenden Gründen erschwert:

- Der Suchraum ist von sehr hoher Dimensionalität.
- Die zu optimierenden Parameter sind stark miteinander verkoppelt.
- Die zu optimierenden Parameter sind beschränkt.
- Die zu optimierende Gütefunktion – die Erkennungsleistung des visuellen Systems – ist nichtlinear und nicht differenzierbar bezüglich der Systemnichtlinearitäten und Kombinationsmerkmale.
- Die Gütefunktion verfügt über Plateaus, was den Einsatz eines Gradientenverfahrens erheblich erschwert[2].
- Die Gütefunktion verfügt über eine große Anzahl von lokalen Optima[3].

Die optimale Bestimmung der Nichtlinearitäten und Kombinationsparameter stellt ein globales Optimierungsproblem dar. Globale Optimierungsmethoden lassen sich in *indirekte* oder *analytische* und *direkte* oder *numerische Methoden* unterscheiden. Während die direkten Methoden sich iterativ dem globalen Optimum nähern, versuchen die indirekten Methoden aus der Analyse der zu optimierenden Funktion das Optimum in einem Schritt zu erreichen. Dabei benötigen die indirekten Methoden im Allgemeinen eine differenzierbare Gütefunktion. Die direkten Methoden kommen hingegen ohne eine analytisch differenzierbare Gütefunktion aus und empfehlen sich daher für das vorliegende Problem. Die direkten Methoden lassen sich in sogenannte *Hill-Climbing* Verfahren und *stochastische Methoden* unterteilen. Während die Hill-Climbing Verfahren im Allgemeinen die Iterationen zum Optimum deterministisch – basierend auf Gradienteninformationen – durchführen, werden die iterativen Schritte der stochastischen Methoden nach Zufallsprinzipien ausgeführt. Verfügt die zu optimierende Funktion über eine große Anzahl von lokalen Optima, so ist der Einsatz von gradientenbasierten Verfahren problematisch. Daher bietet sich in dem vorliegenden Fall in besonderer Weise die Klasse der stochastischen Optimierungsmethoden an, da diese in solchen Fällen weiterhin gute Konvergenzergebnisse zeigen. Die stochastischen Methoden sind im Allgemeinen sehr rechenaufwendig, da sie eine große Anzahl von Gütefunktionsaufrufen

[2] So existieren beispielsweise Merkmale, die sich bezüglich der Erkennungsrate des visuellen Systems neutral verhalten.

[3] Das ergibt sich einerseits aus den zuvor genannten Punkten und wurde andererseits auch schon bei bisherigen Untersuchungen [49, 50] deutlich.

benötigen. Aus diesem Grunde empfiehlt sich innerhalb der stochastischen Methoden in besonderer Weise die Klasse der *populationsbasierten Methoden*, denn diese Algorithmen eignen sich besonders gut für eine Berechnung auf einer parallel verteilten Rechnerarchitektur. Die Gütefunktion im vorliegenden Fall ist die Erkennungsleistung des visuellen Systems auf einer Bilddatenbank vieler Objekte. Die Bestimmung dieser Erkennungsleistung ist aus diesem Grunde rechenaufwendig und empfiehlt daher den Einsatz von populationsbasierten Methoden. Unter diesen Methoden haben sich insbesondere die Evolutionären Algorithmen als leistungsfähig erwiesen [39]. Das vorliegende Optimierungsproblem zielt auf die Bestimmung von kontinuierlichen Werten und nicht auf die Lösung beispielsweise eines kombinatorischen Problems. Bei dieser Art von Problemen haben die *Evolutionsstrategien* im Allgemeinen eine bessere Performanz. Daher werden diese auch in der vorliegenden Arbeit zur Lösung des globalen Entwurfs- bzw. Optimierungsproblems verwendet.

2.2.1 Optimierung von optischen Objekterkennungsmethoden

In den letzten Jahren ist der Einsatz von künstlichen neuronalen Netzen zur Bildverarbeitung stark angestiegen (siehe [7] für einen Überblick). Die Objekterkennung stellt hierbei eine der wichtigsten Fragestellungen dar. Die Erkennung der präsentierten Objektansichten soll möglichst unabhängig von der Translation, der Skalierung und der Rotation des Inputstimulus sein. Ein Schlüsselproblem hierbei ist die Entwicklung von robusten Methoden zur Extraktion und zur Auswahl von invarianten Bildmerkmalen. Die Einbeziehung von Vorwissen ist in dieser Frage von fundamentaler Bedeutung [7]. Das Vorwissen kann dazu genutzt werden, den Netzen eine optimale und problemangepasste Architektur zu geben. Genau dieses wird in der vorliegenden Arbeit dadurch bewerkstelligt, dass gezielt Strukturelemente des visuellen Systems des Menschen eingebracht werden. Jedoch bleiben weiterhin viele Freiheitsgrade des Systems unbestimmt.

Leistungsfähige Methoden für den Entwurf allgemeiner künstlicher neuronaler Netze bieten die Evolutionären Algorithmen. Einen Überblick über die große Menge an Arbeiten in diesem Bereich bietet Yao [52]. Es stellt sich jedoch heraus, dass nur ein sehr kleiner Teil der evolutionär optimierten Netze visuelle Objekterkennungsaufgaben zur Anwendung hat. Der überwiegende Teil der neuronalen Netze wird nicht im Bereich der Objekterkennung eingesetzt. Daher sind die verwendeten Architekturen in der Regel deutlich kleiner als die Netze, die in der visuellen Verarbeitung zur Anwendung kommen. Diese kleineren Netze lassen sich unter der Zuhilfenahme einer sogenannten *Verbindungsmatrix* in ihrer Struktur optimieren. Dazu wird im Allgemeinen eine obere Dreiecksmatrix aufgestellt, die die Verbindungen aller Neuronen untereinander direkt kodiert. Im Rahmen der evolutionären Suche wird dann ermittelt, wo eine Verbindung besteht und wo nicht. Die Bestimmung der einzelnen

Verbindungsgewichte der üblicherweise vorwärtsgerichteten Netze erfolgt zum einen mit Hilfe eines gesonderten Lernalgorithmus. Hierzu wird in einer großen Zahl der Fälle gradientenbasiertes Lernen eingesetzt. Zum anderen wird die Bestimmung der Netzgewichte in die evolutionäre Suche nach der Verbindungsstruktur mit integriert.

Neuronale Netze, die zur Lösung von visuellen Objekterkennungsaufgaben eingesetzt werden, benötigen im Allgemeinen eine sehr große Anzahl von Neuronen. Um die Anzahl der freien Netzparameter jedoch trotzdem möglichst gering und damit für eine Optimierung noch handhabbar zu halten, werden in dieser Arbeit biologisch inspirierte Randbedingungen mit in den Aufbau der Netze eingebracht.

Eine in diesem Zusammenhang sehr häufig genutzte hierarchische Struktur ist die des Neocognitrons [8], die auch in zahlreichen Abwandlungen zum Einsatz kommt. Einige wenige Wissenschaftler setzen evolutionäre Methoden ein, um diese hierarchischen Netze zu optimieren. Pan et al. [27] beispielsweise nutzen eine evolutionäre Optimierung, um Merkmale in mittleren Schichten der visuellen Hierarchie zu optimieren. Die verwendeten Zwischenzielmuster, auf die die Merkmale hin optimiert werden, müssen jedoch manuell erzeugt werden. Shi et al. [44] hingegen setzen Genetische Algorithmen ein, um diese Zwischenzielmuster zu generieren. Zur Bestimmung der Merkmale des Netzes werden die konventionellen überwachten Methoden des Neocognitrons verwendet [8]. Honavar und Uhr [16] bilden, gesteuert durch die Fehlklassifikation des Erkennungsnetzes, eine Menge von optischen Merkmalen. Als Kandidaten für die neuen Merkmale werden Ausschnitte aus Zwischenschichten des Erkenners verwendet, die eine hohe Aktivierung aufweisen. Nach der Erzeugung der neuen Merkmale ist das Netz wieder mit Hilfe eines Gradientenabstiegsverfahrens neu zu trainieren. Die Aufgabe des Netzes ist die Erkennung von einfachen Linienzeichnungen. Teo und Sim [46] verwenden Design-of-Experiment(DoE)- und Orthogonal-Array(OA)-Methoden, um eine Reihe von freien Parametern innerhalb des angewendeten Neocognitrons zu optimieren. In [45] vergleichen sie diese Optimierung mit einer evolutionären Optimierung, bei der sie sich jedoch auf die Optimierung weniger Netzparameter beschränken. Die Aufgabe des Netzes war die Klassifikation von 10 handgeschriebenen Ziffern.

Zusammenfassend kann gesagt werden, dass alle Arbeiten, die sich mit der evolutionären Optimierung von Merkmalen in biologisch inspirierten Objekterkennern beschäftigen, nur relativ einfache Anwendungen, wie z.B. Ziffern- oder Buchstabenerkennung, zum Ziel haben. Weiter müssen häufig Teile des Erkenners mit Hilfe von manuellen Einstellungen nachoptimiert werden. Auch wird in diesen Arbeiten nicht darauf eingegangen, wie gut die optimierten Erkennungssysteme, bzw. die optimierten Merkmale, sich zur Erkennung von Objekten anderer Domänen eignen. Im folgenden Kapitel wird eine evolutionäre Optimierung eines biologisch inspirierten hierarchischen Objekterkenners vorgestellt, die ohne jegliche manuelle Nacheinstellungen auskommt und zu-

dem eine anspruchsvolle Erkennung von realen 3D-Objekten zur Aufgabe hat. Die gefundenen Erkenner werden zusammen mit ihren optimierten komplexen Merkmalen auf ihre Verallgemeinerbarkeit hin überprüft.

2.2.2 Systementwurf mit Evolutionsstrategien

Evolutionäre Methoden als technisches Entwurfskonzept haben in zahlreichen Anwendungen erfolgreich gezeigt, komplizierte neuronale Strukturen optimieren zu können [52]. Sie bieten sich – wie bereits weiter oben erwähnt – in besonderer Weise dazu an, das vorliegende Entwurfsproblem zu lösen. Nachfolgend werden Evolutionäre Algorithmen kurz im Allgemeinen und die Evolutionsstrategien im Besonderen eingeführt.

Die Grundidee der evolutionären Optimierung liegt darin, das fundamentale Anpassungsprinzip der belebten Natur, die Evolution, auf den technischen Bereich zu übertragen und hier zur Optimierung vielfältigster Problemstellungen zu nutzen. Die Evolutionsstrategien (ES) und die Genetischen Algorithmen (GA) wurden Ende der 70er Jahre von Rechenberg und Holland vorgeschlagen. Schon die Übertragung weniger evolutionärer Prinzipien genügt, um eine leistungsstarke evolutionäre Optimierung zu implementieren. Zu Evolutionären Algorithmen zählt man eine große Menge von unterschiedlichen Algorithmen, die man grob in die folgenden Richtungen einteilen kann: evolutionäres Programmieren (EP), Evolutionsstrategien (ES), Genetische Algorithmen (GA) und genetisches Programmieren (GP). Da die Grenzen zwischen diesen Strömungen eher fließend sind und z.T. auch viele Mischformen anzutreffen sind, ist eine genaue Zuordnung weder möglich noch sinnvoll. Im Folgenden werden die Grundprinzipien und die Terminologie vorgestellt, die allen evolutionären Optimierungsalgorithmen im Wesentlichen zugrunde liegen.

Die angewandten Methoden lehnen sich in Funktion und Bezeichnung eng an das biologische Vorbild an, erheben aber im Allgemeinen keinen Anspruch auf biologische Korrektheit. Sie orientieren sich vielmehr daran, wie die statistischen Optimierungsalgorithmen effizient – meist in Bezug auf spezifische Problemstellungen – gestaltet werden können. Eine Art von Problemen, bei denen der Einsatz evolutionärer Verfahren besonders sinnvoll und erfolgversprechend ist, ist die folgende: Die Problemstellungen sind nicht oder nur mit sehr hohem Aufwand analytisch zu lösen, aber gleichzeitig kann ein formal zulässiger Lösungsvorschlag quantitativ in Bezug auf seine Güte bewertet werden.

Der systematische Ablauf der evolutionären Optimierung ist in Abbildung 2.5 dargestellt. Zu Beginn werden zufällig oder unter Ausnutzung von Vorwissen zulässige Lösungen des gesuchten Problems, die sogenannten *Individuen*, initialisiert. Die ein Individuum definierenden Informationen werden in jeweils einem *Genom* kodiert. Man spricht hierbei auch von dem *Genotypen* eines Individuums. Das Genom wird weiter unterteilt in eine Anzahl von *Chromo-*

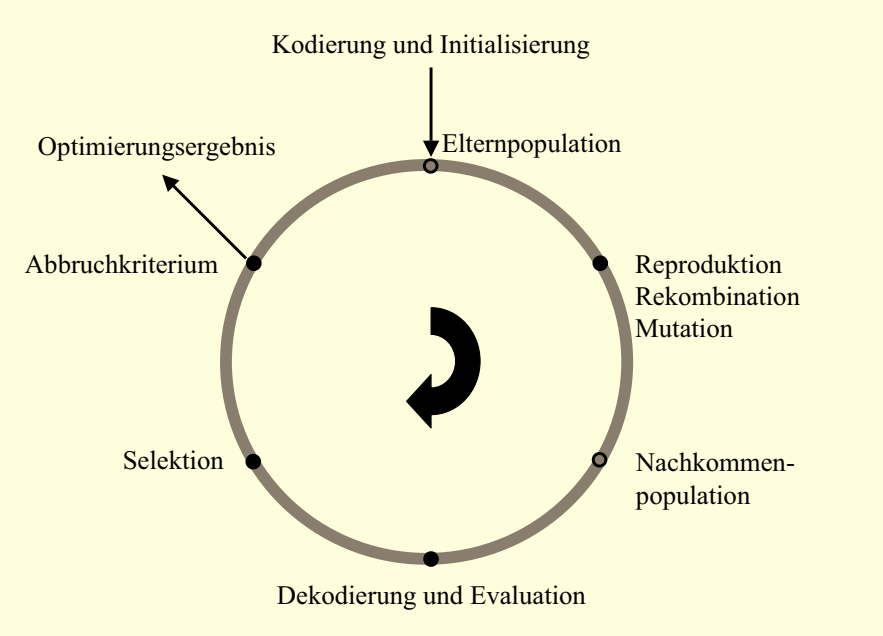

Abbildung 2.5: Schematische Darstellung der evolutionären Optimierungsschleife.

somen, die wiederum aus einer Anzahl von *Allelen* bestehen. Die Menge dieser zu Beginn initialisierten Individuen stellt die *Population* der *Eltern* dar. Aus dieser Population wird dann, mittels *Reproduktion* und eventueller *Rekombination* und *Mutation*, die Population der *Nachkommen* gebildet. In diesem Prozess werden die im Genom kodierten Informationen an die nächste *Generation* übertragen. Hierbei sorgt der Rekombinationsoperator im Allgemeinen dafür, dass die Eigenschaften von zwei oder mehreren Eltern bei der Erzeugung eines Nachkommens kombiniert werden. Der Mutationsoperator sorgt dafür, dass die kodierten Eigenschaften eines Genoms zufällig variiert werden. Nachdem die Nachkommenpopulation in dieser Art gebildet wurde (Reproduktion) wird sie dekodiert. Das bedeutet der *Phänotyp*, die interessierende Lösung des Problems (beispielsweise ein optimal geformtes Turbinenblatt), wird aufgebaut. Die Eigenschaften eines jeden Phänotyps sind in seinem Genotypen kodiert. Die dazu notwendige Zuordnung von einem Genotypen zu dem entsprechenden Phänotypen ist durch die sogenannte *Genotyp-Phänotyp-Abbildung* (GPA) definiert. Dieser Abbildung kommt eine Schlüsselposition innerhalb des gesamten Formalismus zu. In einem nächsten Schritt wird die Güte jedes aufgebauten Phänotyps bestimmt und als *Fitness* dem jeweiligen Individuum zugeordnet. Anhand dieser Fitness wird nun eine Rangfolge der Individuen im Hinblick auf die vorliegende Evaluierungs- oder *Fitnessfunktion* bestimmt. Diese Rangfolge ist für die nun folgende Auswahl oder *Selektion* entscheidend. Das bei der Selektion angewendete Grundprinzip lautet: Die Individuen mit der höchsten Fitness setzen sich durch. Das kann auf zwei Arten realisiert werden. Bei

der *deterministischen* Selektion werden die Individuen mit der größten Fitness automatisch die Eltern der nächsten Generation. Bei der *stochastischen* Selektion hingegen erhöht die höhere Fitness eines Individuums lediglich die Wahrscheinlichkeit, in die Elternpopulation aufgenommen zu werden.

Entscheidend bei der Selektion ist auch, dass ein gewisser *Selektionsdruck* herrscht. Dieser ergibt sich aus dem Verhältnis der Auswahlwahrscheinlichkeit des besten Individuums zur durchschnittlichen Auswahlwahrscheinlichkeit aller Individuen des Auswahlpools[4]. Mit Hilfe der Selektion wurden also diejenigen Individuen der Nachkommenpopulation ausgewählt, die die Elternpopulation der nächsten Generation stellen werden. Damit ist der Kreislauf geschlossen, d.h. eine Generation innerhalb des iterativen Prozesses der evolutionären Optimierung durchlaufen. Nach Durchlauf vieler dieser Generationen verbessert sich im Allgemeinen die Fitness der Individuen immer weiter. Der Prozess wird dann gestoppt, wenn entweder eine Lösung gefunden wurde, die eine geforderte Güte aufweist, oder der Algorithmus ein anderes Abbruchkriterium erreicht, wie beispielsweise der Ablauf einer vorher festgelegten Verarbeitungszeit.

In der vorliegenden Arbeit werden Evolutionäre Algorithmen verwandt, die zu der Gruppe der Evolutionsstrategien zählen. Im folgenden Abschnitt wird daher auf die Besonderheiten dieser Richtung der Evolutionären Algorithmen eingegangen. Außerdem werden die gerade erläuterten Grundbegriffe mathematisch formaler beschrieben.

Evolutionsstrategien

Evolutionsstrategien (ES) wurden von Ingo Rechenberg [29] vorgeschlagen. Ein allgemeines Merkmal dieser Richtung der Evolutionären Algorithmen ist die stärkere Betonung der Variation durch Mutations- anstatt durch Rekombinationsoperatoren.

Die phänotypischen Eigenschaften eines Individuums sind in dem *Objektparametervektor* $\mathbf{x} = (x_1, \ldots, x_n)^\top$ definiert und liegen hier meist in der Form von Fließkommazahlen vor. Ein typischer Mutationsoperator ist folgendermaßen definiert:

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}(t-1) + \mathbf{z} \\ z_i &\sim N(0, \sigma_{gs}^2). \end{aligned} \tag{2.10}$$

Hierbei ergibt sich der Objektparametervektor in der Generation t, $\mathbf{x}(t)$ aus der Addition von dem Objektparametervektor der Generation $t-1$, $\mathbf{x}(t-1)$ und einem Zufallsvektor $\mathbf{z}$, dessen Komponenten z_i normal verteilte Zufallszahlen mit dem Mittelwert 0 und der Standardabweichung σ_{gs} enthalten. Dies sei durch die Schreibweise $z_i \sim N(0, \sigma_{gs}^2)$ dargestellt. Hierbei beschreibt $N(\mu, \sigma^2)$

[4]Der Auswahlpool ist nicht notwendigerweise die Nachkommengeneration, sondern kann auch aus der Vereinigung der Nachkommengeneration und der Elterngeneration bestehen oder auf vielfältige andere Weisen gebildet werden.

eine Gaußsche Normalverteilung mit einem Mittelwert von μ und einer Standardabweichung von σ. Die Werte z_i werden – dargestellt durch das Symbol „$\sim$" – aus dieser Verteilung gezogen. Die Stärke der Mutation ist durch σ_{gs} festgelegt. Aus diesem Grund wird σ_{gs} auch als *Schrittweite* bezeichnet. Eine große Bedeutung kommt der Einstellung dieses Wertes zu. Bei einem zu geringen Wert für σ_{gs} dauert die evolutionäre Optimierung unnötig lange. Außerdem sinkt auch die Wahrscheinlichkeit, mit der Individuen aus lokalen Minima im Fitnessraum entkommen können. Bei einem zu hohen Wert für σ_{gs} hingegen wird die Konvergenz der Objektparameter zu optimalen Werten stark erschwert. Somit kommt der Wahl der Schrittweite eine zentrale Bedeutung zu, die in erheblichem Maße über die Performanz einer Optimierung entscheidet. Hinzu kommt, dass sich im Allgemeinen der beste Wert für die Schrittweite innerhalb eines Optimierungslaufes ändert. So sind typischerweise zu Beginn einer Optimierung große Schrittweiten sinnvoll, bis die Individuen eine erste Grobeinstellung der Parameterwerte vorgenommen haben. Dann hingegen ist meist eine kleinere Schrittweite gut geeignet, um die abschließende Feineinstellung der Objektparameterwerte zu ermöglichen.

Ein Kennzeichen der Evolutionsstrategien ist es, dass die Schrittweite nicht fest vorgegeben, sondern variabel gehalten und darüber hinaus für jedes Individuum individuell festgelegt wird. Die Einstellung des Wertes wird durch den Mechanismus der *Selbstadaptation* gesteuert. Hierzu wird die Schrittweite, die in diesem Zusammenhang auch als *Strategieparameter* bezeichnet wird, zusammen mit dem Objektparametervektor im Genom kodiert und ist hier auch Variationen durch Mutation und Rekombination unterworfen. Die Objektparameter werden unter Verwendung der im eigenen Genom abgelegten Schrittweiten mutiert. Auf diese Weise werden durch die Evolution nicht nur die Objektparameter, sondern auch die Schrittweiten auf einen optimalen Wert eingestellt. Anders jedoch als die Objektparameter, deren Werte einen direkten Einfluss auf die Fitness eines Individuums haben, ist der Einfluss der Schrittweite auf die Fitness ein indirekter. So ist anzunehmen, dass nur die Individuen, die auch eine geeignete Schrittweite verwenden, eine gute Fitness erlangen. Durch eine hohe Fitness der Individuen wird somit auch eine hohe Wahrscheinlichkeit zur Ausbreitung einer günstigen Schrittweite sichergestellt. Dieser zugrunde liegende Prozess wird auch als *Selektion zweiter Ordnung* bezeichnet, siehe hierzu z.B. Schwefel [41].

Die Anzahl der in der Elternpopulation vorhandenen Individuen wird im Allgemeinen bei ES mit μ, und in der Nachkommengeneration mit λ bezeichnet. Bei der Selektion der Individuen für die Elternpopulation der nächsten Generation gibt es zwei unterschiedliche Strategien. In der ersten, der sogenannten *Kommastrategie*, besteht der Auswahlpool, aus dem die μ Individuen für die nächste Elterngeneration aufgrund ihrer Fitness selektiert werden, ausschließlich aus den λ Individuen der Nachkommenpopulation. Diese Strategie wird formal mit dem Ausdruck (μ, λ) bezeichnet. Bei der zweiten Strategie, der

sogenannten *Plusstrategie*, besteht der Auswahlpool nicht nur aus den Individuen der Nachkommenpopulation, sondern auch aus Individuen der momentanen Elternpopulation. Werden beispielsweise alle Eltern mit in den Auswahlpool mit einbezogen, so wird diese Strategie mit $(\mu + \lambda)$ bezeichnet.

Der Vorteil der Plusstrategie besteht darin, dass eine einmal gefundene sehr gute Lösung des Problems nicht wieder innerhalb der Evolution verloren gehen kann und somit zumindest das beste Individuum einer Generation im Verlauf der Optimierung sich niemals in der Fitness verschlechtern kann. Der Nachteil davon jedoch ist, dass dadurch die Tendenz der Optimierung, in lokalen Maxima der Fitnesslandschaft zu verharren, ansteigt. Diese Gefahr erweist sich bei der Kommastrategie als geringer. Um allerdings auch hier nicht eine einmal gefundene, sehr gute Lösung des Optimierungsproblems zu verlieren, wird in der Regel das beste jemals evaluierte Individuum fortwährend gespeichert, aber nicht in eine Population integriert.

Für die Auswahl der Individuen für die nächste Elterngeneration existiert eine Vielzahl von unterschiedlichen Selektionsoperatoren [1], auf die hier jedoch nicht weiter eingegangen werden soll. Im Folgenden wird stets die leistungsfähige und stark verbreitete Methode der sogenannten *deterministischen Selektion* verwendet. Hierbei wird die Auswahl allein an der Fitness der Individuen festgemacht. Sind also μ Individuen aus dem Auswahlpool zu selektieren, so werden die μ Individuen mit der besten Fitness ausgewählt. Bei Fitnessgleichheit von Individuen entscheidet der Zufall.

Mutation mit Schrittweitenadaptation

Die oben beschriebene Methode der Adaptation der Schrittweiten wird als globale Schrittweitenadaptation bezeichnet, da *eine* Schrittweite σ_{gs} die Mutation *aller* Einträge des Objektparametervektors steuert. Formal lässt sich die Mutation von $\mathbf{x}$ und σ_{gs} jetzt folgendermaßen formulieren:

$$\begin{aligned} \sigma_{gs}(t) &= \sigma_{gs}(t-1)\exp(\tau_0\, z') && (2.11)\\ \mathbf{x}(t) &= \mathbf{x}(t-1) + \mathbf{z} && (2.12)\\ z' &\sim N(0,1),\ z_i \sim N(0, \sigma_{gs}(t)^2). \end{aligned}$$

Hierbei kontrolliert der Parameter τ_0 die Mutation der Schrittweite σ_{gs}. Im Gegensatz zu Gleichung (2.10) wird jetzt die Mutation von $\mathbf{x}$ über eine variable Schrittweite $\sigma_{gs}(t)$ gesteuert. Die zur Mutation eines Individuums verwendete Schrittweite $\sigma_{gs}(t)$ wird aus dem jeweiligen Genom des Individuums ausgelesen. Die Schrittweite selbst unterliegt in jeder Generation einer zufälligen Mutation gegeben durch Gleichung (2.11). Zur Steuerung dieser Mutationsstärke ist jetzt erneut der Parameter τ_0 festzulegen. Jedoch sind die Auswirkungen dieses Wertes auf die Leistungsfähigkeit der Optimierung von weitaus geringerer Bedeutung. Einfluss auf die Wahl von τ_0 hat die Dimension des Suchraumes, die

gleich der Dimension des Objektparametervektors n ist. Eine gute Heuristik für τ_0 ist $\tau_0 \sim \frac{1}{\sqrt{n}}$ [40].

Durch die global geregelte Einstellung der Schrittweite für die Mutation aller Komponenten des Objektparametervektors ist eine differenzierte Behandlung der unterschiedlichen Dimensionen des Genotypraumes nicht möglich. Es kann jedoch sein, dass unterschiedliche Objektparameter stark unterschiedliche Schrittweiten erfordern. Als eine Möglichkeit, diesem Problem zu begegnen, bietet sich die Methode der individuellen Schrittweitenadaptation an.

Anders als bei der globalen Schrittweitenadaptation wird jetzt für jeden Objektparameter eine individuelle Schrittweite festgelegt. Diese sind in dem Schrittweitenvektor $\boldsymbol{\sigma}_{\mathrm{i}s} = (\sigma_{\mathrm{is},1}, \ldots, \sigma_{\mathrm{is},n})^\top$ abgelegt. Die Gleichungen (2.11) und (2.12) modifizieren sich damit zu:

$$\begin{aligned} \sigma_{\mathrm{i}s,i}(t) &= \sigma_{\mathrm{i}s,i}(t-1)\exp(\tau\, z' + \eta\, z_i'') && (2.13)\\ \mathbf{x}(t) &= \mathbf{x}(t-1) + \mathbf{z} && (2.14)\\ z' &\sim N(0,1),\ z_i'' \sim N(0,1),\ z_i \sim N(0, \sigma_{\mathrm{is},i}(t)^2). \end{aligned}$$

Bei der Mutation von $\boldsymbol{\sigma}_{\mathrm{i}s}$ wird mit zwei Mutationsanteilen gearbeitet: einem weiterhin globalen und einem individuellen. So ist z' ein skalarer Zufallswert, der die Mutationsrichtung des gesamten Vektors $\boldsymbol{\sigma}_{\mathrm{i}s}$ beeinflusst, während z_i'' die i-te Komponente des Zufallsvektors $\mathbf{z}''$ ist, der jede Komponente von $\boldsymbol{\sigma}_{\mathrm{i}s}$ individuell mutiert. Über die Parameter τ und η ist es möglich, die Geschwindigkeit der jeweiligen Adaption zu steuern. Eine gute heuristische Wahl[5] für diese Parameter ist $\tau = \frac{1}{\sqrt{2n}}, \eta = \frac{1}{\sqrt{2\sqrt{n}}}$ [40], mit n gleich der Anzahl der Objektparameter.

Mit der individuellen Schrittweitenadaption steigt die Anzahl der Freiheitsgrade für die evolutionäre Optimierung, da jetzt zusätzlich zu den n Objektparametern noch n Strategieparameter (die Schrittweiten) anzupassen sind. Das erweist sich als nicht unproblematisch.

Durch die Einführung des Verfahrens der globalen Schrittweitenadaptation wird es möglich, die Stärke der Mutation mit der Zeit zu variieren. Die Steuerung erfolgt hierbei durch die Veränderung der Standardabweichung der normalverteilten Zufallszahlen (vgl. Gleichung (2.11)). Mit der Methode der individuellen Schrittweitenadaptation wird es möglich, die Stärke der Mutation bezüglich jedes Objektparameters getrennt zu variieren (vgl. Gleichung (2.13)). Auf diese Weise wird die Gaußsche Hyperkugel, die die Stärke der Mutation steuert, zu einem Gaußschen Hyperellipsoiden. Berücksichtigt man weiter, dass die Objektparameter im Allgemeinen auch miteinander korreliert sind, so wäre eine Drehung dieses Mutationshyperellipsoiden wünschenswert. Hierzu existieren unterschiedliche Verfahren. Eine interessante Methode stellt das Verfahren der Kovarianzmatrixadaptation (CMA)[11, 12] dar. In Abbildung 2.6 ist eine

[5] Es sei bemerkt, dass abhängig von der jeweiligen Topologie des Optimierungsproblems eine andere Wahl dieser Größen besser geeignet sein kann.

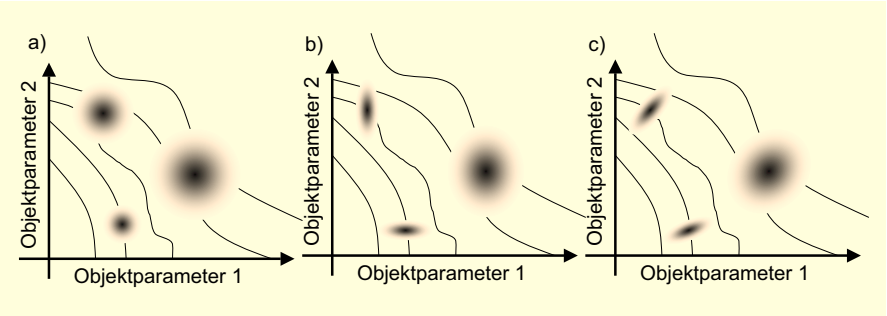

Abbildung 2.6: Schematische Darstellung möglicher Verteilungswahrscheinlichkeiten von Nachkommen im Fitnessraum (ausgehend von drei Eltern) für a) globale Schrittweitenadaptation b) individuelle Schrittweitenadaptation und c) Kovarianzmatrixadaptation (CMA). Die eingezeichneten Linien stellen die Höhenlinien der zu optimierenden Fitnessfunktion dar.

schematische Darstellung für mögliche Verteilungen der Nachkommen (ausgehend von drei Eltern) für die unterschiedlichen Formen der Schrittweitenadaptation gegeben. Die CMA ist in der Lage, die Konvergenzgeschwindigkeit bei der Optimierung bestimmter Probleme weiter zu erhöhen. Der Nachteil dieses Verfahrens liegt jedoch in der größeren Anfälligkeit, in suboptimalen Lösungen zu enden.

Rekombination

Genau wie es bei den Mutationsoperatoren eine große Zahl von Varianten gibt, so ist auch die Variationsbreite der Rekombinationsoperatoren groß. Gerade auch bei den Rekombinationsoperatoren kann es sinnvoll sein, problemspezifische Operatoren zu entwerfen. Wenn sich eine Gesamtlösung in eine Anzahl von Unterlösungen aufteilen lässt, sollten bei einer Rekombination diese Teile als eine Einheit betrachtet und entsprechend rekombiniert werden.

An dieser Stelle sollen zwei einfache Standardrekombinationsverfahren kurz dargestellt werden. Die *diskrete Rekombination* geht im Allgemeinen von zwei Individuen mit Parametervektoren gleicher Dimension aus. Aus diesen beiden Vektoren wird ein neuer Parametervektor erzeugt, wobei der Wert jeder Komponente nun mit gleicher Wahrscheinlichkeit aus der entsprechenden Komponente einer der beiden Eingangsvektoren stammt. Bei der *generalisierten intermediären Rekombination* hingegen wird der Wert der Komponente des neuen Vektors zufällig gleichverteilt aus dem Intervall gezogen, das durch die Werte der Eingangsvektorkomponenten aufgespannt wird.

Während die Variationen durch Mutationsoperatoren theoretisch zu jedem beliebigen Wert in einer Vektorkomponente führen können, wird bei vielen Rekombinationsoperatoren die Variation auf den Raum der schon gefundenen Komponenten beschränkt. Auf diese Weise werden neue Kombinationen

von Komponenten als neue Gesamtlösungen generiert. Dieser Vorgehensweise liegt die Idee zugrunde, dass zumindest eine gewisse Anzahl von „Teillösungen" für bestimmte Komponenten existiert. Diese von Individuen schon erfolgreich gefundenen Teillösungen werden durch die Rekombination unter Umständen nicht zerstört, sondern zusammen mit anderen Teillösungen in einem neu generierten Individuum genutzt.

Genotyp-Phänotyp-Abbildung

Eine wichtige Stellung innerhalb der evolutionären Optimierung nimmt die Genotyp-Phänotyp-Abbildung (GPA) ein (In Abbildung 2.5 fällt die Verwendung der GPA unter den Begriff „Kodierung" bzw. „Dekodierung".)[6]. In ihr wird der Übergang vom Genotypraum zum Phänotypraum festgelegt, also von dem Raum, in dem die Variationsoperatoren wie Mutation und Rekombination angewendet werden, zu dem Raum, in dem die Fitnessevaluierungen stattfinden. Der Ausdruck „Abbildung" soll hier nicht im mathematisch strengen Sinne verwendet werden, sondern vielmehr als eine Festlegung von Regeln verstanden werden, die den Übergang von dem einen in den anderen Raum beschreiben.

Die Definition des Phänotyps ist u.U. problemabhängig. Die Frage, die sich ergibt, ist die: Was gehört noch zur Genotyp-Phänotyp-Abbildung und was bereits zur Fitnessevaluierung? Bei einer evolutionären Optimierung von künstlichen neuronalen Netzen beispielsweise, bei der die Struktur der Verschaltung optimiert wird, um eine bestimmte Klassifikationsaufgabe besonders gut lösen zu können, kann die nachfolgende Einstellung der Netzgewichte mit einem Backpropagation-Algorithmus einerseits noch als ein Teil der GPA angesehen werden. In diesem Falle ist der Phänotyp das fertige Netz, das die Klassifikation direkt ausführen kann. Andererseits kann man auch die Aufgabe der Evolution darin sehen, eine für die Klassifikationsaufgabe geeignete neuronale Verschaltungsstruktur zu finden. In diesem Fall gehört das anschließende Training zum Prozess der Fitnessbestimmung. Einen tatsächlichen Unterschied gibt es in diesem Beispiel nicht. In anderen Fällen ist die Festlegung des Phänotyps eindeutiger. Ein Beispiel hierfür ist die Parameteroptimierung einer mathematischen Formel, bei der die Werte einer festgelegten Menge von Parametern so zu bestimmen sind, dass eine vorgegebene Menge von Ein- und Ausgangsdaten den kleinsten quadratischen Fehler einnehmen. In diesem Fall stellt offensichtlich die Formel mit den Parameterwerten den Phänotyp dar.

Bei der Festlegung, was noch Genotyp und was bereits Teil der Genotyp-Phänotyp-Abbildung ist, soll Folgendes gelten: Die kodierten Systemeigenschaften, die Gegenstand von Mutation und Rekombination sein können, beschreiben den Genotypen. Alle festen Einstellungen, die innerhalb der evolu-

[6] Häufig wird auch der Begriff „Repräsentation" synonym zu „Kodierung" verwendet. Um jedoch Verwechselungen zu vermeiden, wird in dieser Arbeit der Begriff der Repräsentation ausschließlich im Zusammenhang mit der Repräsentation von visuellen Objekten verwendet.

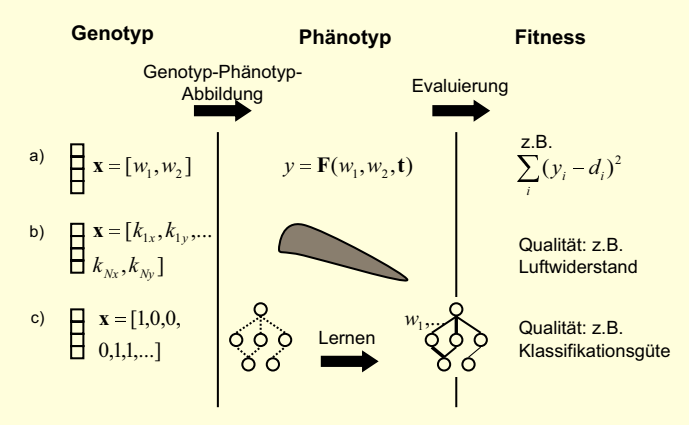

Abbildung 2.7: Schematische Darstellung der Beziehungen von Genotyp, Phänotyp und Fitness anhand von drei Beispielen. Evolutionäre Optimierung von a) Funktionsparametern, b) eines aerodynamischen Flügels und c) eines neuronalen Netzes.

tionären Optimierung nicht weiter verändert werden und die den Aufbau des Phänotyps bestimmen, zählen zur GPA. Betrachtet man z.B. die evolutionäre Formoptimierung eines aerodynamischen Flügels, so beinhaltet der Genotyp möglicherweise die Koordinatenwerte von einer Anzahl von Knotenpunkten einer Spline-Kodierung des Flügels. Die Berechnung der Splines aus den Knotenpunkten ist bereits Teil der Genotyp-Phänotyp-Abbildung. In Abbildung 2.7 sind die erläuterten Beziehungen zwischen Genotyp, Phänotyp und Fitness noch einmal beispielhaft schematisch dargestellt. Im Beispiel (a) werden die Funktionsparameter so optimiert, dass z.B. der quadratische Fehler aus Funktionswerten y_i und Zielwerten d_i, mit $i = 1, ..., n$ minimal ist. Im Falle der Tragflügeloptimierung (b) besteht der Genotyp aus den Knotenpunkten, aus denen mit Hilfe einer Spline-Kodierung eine dreidimensionale Struktur gewonnen wird. Im Falle der Netzoptimierung (c) kann man als Phänotyp entweder die Netzverbindungsstruktur sehen oder aber das Netz mit den verwendeten Netzgewichten $w_1, w_2, ..., w_n$. In der ersten Sichtweise zählt der Lernvorgang des Netzes zu dem Prozess der Evaluation. In der zweiten Sichtweise gehört das Lernen zur Genotyp-Phänotyp-Abbildung. In dieser Arbeit wird in diesem Zusammenhang auch von einer indirekten Kodierung gesprochen. Von einer direkten Kodierung spricht man, wenn alle phänotypischen Eigenschaften explizit im Genotyp kodiert sind. Auf die Kopplung von Evolution und lokalem Lernen und auf die Bedeutung der Genotyp-Phänotyp-Abbildung wird noch in Abschnitt 2.3 näher eingegangen.

2.3 Kopplung von Evolution und lokalem Lernen

Die Entwicklung des visuellen Systems des Menschen wird von den zwei eng miteinander verbundenen Entwicklungsprozessen *Phylogenese* und *Ontogenese* bestimmt. Die Phylogenese beschreibt die Entwicklung innerhalb der biologischen Evolution des Menschen über alle Generationen hinweg. Diese Entwicklung nahm ihren Anfang bei den ersten Einzellern und dauert bis heute an. Die Ontogenese beschreibt die individuelle Entwicklung des visuellen Systems eines einzelnen Menschen beginnend von dem Heranwachsen einer befruchteten Eizelle bis hin zum erwachsenen Menschen[7]. Beide Entwicklungsprozesse – Phylogenese und Ontogenese – sind sehr eng miteinander verwoben und beeinflussen sich gegenseitig. Die ontogenetische Entwicklung des menschlichen visuellen Kortex ist nicht nur als die bloße Ausführung eines genetisch festgelegten Bauplanes anzusehen. Vielmehr verläuft die Entwicklung in engem Zusammenspiel mit äußeren Reizen. Quartz und Sejnowski führen eine Reihe von Belegen an, dass diese Entwicklung ein gezieltes Wachstum ist und nicht – wie z.T. auch vermutet – ein Absterben von Neuronen nach einer Phase der massiven Überproduktion [28]. Ulyings et al. [47] konnten feststellen, dass sich der visuelle Kortex von Ratten, die in einer an optischen Reizen sehr armen Umgebung aufwuchsen, sehr viel weniger ausbildete als der von Ratten, die im Gegensatz dazu in einer optisch sehr reichhaltigen Umgebung aufwuchsen. Es ist also davon auszugehen, dass im Genom nur eher allgemeine Architekturrichtlinien festgelegt sind und die genaue Vernetzung über reizinduzierte Strukturierungsmechanismen gesteuert wird. Gut vorstellbar ist, dass etwaige Kontrollparameter der Mechanismen genetisch vorgegeben sind. So könnte etwa die zeitliche Dauer des Wachstums einer Kortexschicht determiniert sein, die exakte Vernetzung jedoch würde einem Strukturierungsprozess unterliegen, der seinerseits wieder von der Art seiner sensorischen Eingänge beeinflusst wird. Dieser Strukturierungsprozess ist als eine Form von Lernprozess anzusehen, der unüberwacht oder überwacht ablaufen kann. In der vorgeburtlichen Phase der Ontogenese ist noch kein sensorischer Input für das visuelle System vorhanden, daher verläuft in dieser Zeit die Entwicklung des visuellen Kortex weitgehend[8] ohne Input.

Die im Abschnitt 2.2 dargestellten evolutionären Optimierungsverfahren verwenden in der überwiegenden Zahl der Anwendungen die sogenannte direkte Kodierung. Hierbei werden alle Objektparameter explizit in das Genom kodiert. Durch diese direkte Form der Genotyp-Phänotyp-Abbildung sind die Evolutionsstrategien gezwungen, auf einem sehr hochdimensionalen Suchraum

[7] Über den Endzeitpunkt der Ontogenese gibt es unterschiedliche Meinungen. In manchen Definitionen endet diese Art der Entwicklung mit der Geburt, in anderen hingegen erst mit dem Tod des Lebewesens.

[8] Es konnten auch in dieser Zeit schon vorgeburtliche Aktivierungsmuster nachgewiesen werden [51].

zu arbeiten. Biologisch wahrscheinlicher ist allerdings eine indirekte Form der Kodierung. Bei dieser ist der Genotypraum von einer viel geringeren Dimensionalität als der Phänotypraum[9]. Eine direkte Form der Kodierung scheint auch allein unwahrscheinlich wegen der Tatsache, dass die Informationsmenge, die im menschlichen Genom kodiert werden kann, nicht ausreicht, um die Verbindungen aller Neuronen zueinander im Kortex explizit zu kodieren. Eine Genotyp-Phänotyp-Abbildung könnte, realisiert durch eine feste Abbildungsvorschrift, von einem niedrigdimensionalen Genotypraum auf einen hochdimensionalen Phänotypraum führen.

Eine entsprechende Abbildung bzw. diese Art der indirekten Kodierung ist nicht unproblematisch. Ein Problem ist dadurch gegeben, dass im Allgemeinen nach der Reduktion des Genotypraumes nicht mehr alle Phänotypen im Genom kodierbar sind. Es muss also sichergestellt werden, dass keine potentiell optimalen Phänotypen ausgeschlossen werden, sondern nur solche, bei denen eine geringe Fitness angenommen werden kann oder die in gleicher Weise durch noch darstellbare Phänotypen substituiert werden können. D.h., damit sollten nach Möglichkeit unnötige Redundanzen eliminiert werden, die den Suchprozess nur erschweren. Es kann allerdings nicht generell gesagt werden, dass alle redundanten Phänotypen den evolutionären Suchprozess erschweren. Ein weiteres Problem ist, dass sichergestellt werden muss, dass die sogenannte *starke Kausalität* erhalten bleibt [43]. Dies bedeutet im Wesentlichen, dass die Nachbarschaftsverhältnisse bei der Abbildung vom Genotyp- in den Phänotypraum erhalten bleiben müssen, da sonst die Optimierung stark erschwert wird. Der Hauptvorteil, der sich jedoch durch die Dimensionsreduktion des Genotypraumes ergibt, ist eine in der Regel erleichterte Suche nach dem globalen Optimum.

Ist wie in vielen Fällen die indirekte Kodierung mit einem Lernverfahren gekoppelt, so gewinnt der einzelne Phänotyp, das Individuum, den Vorteil der flexibleren Anpassung an eine sich verändernde Umwelt. In der Natur entwickelt sich ein Individuum während der Ontogenese in Rückkopplung mit der momentanen Umwelt und hat damit die Möglichkeit, sich besser an die momentan herrschende Situation anzupassen. Bei einer rein direkten Kodierung ohne individuelles Lernen können sich die Individuen nicht derart schnell an eine veränderte Umwelt anpassen. Ein Nachteil dieser Art der Kopplung von Evolution und Lernen ist jedoch die verlängerte Entwicklungszeit eines Individuums. Die Tendenz zu längeren Entwicklungszeiten ist in der Natur speziell bei höheren Lebensformen zu finden. Das Pendant in der technischen Umsetzung ist die zusätzliche Rechenzeit, die der Lernprozess jedes Individuums bei der Dekodierung benötigt.

[9]Die Annahme, dass in der Natur der Phänotypraum grundsätzlich größer als der Genotypraum ist, ist nicht notwendigerweise immer der Fall und hängt von der Definition des Phänotypraumes ab. Ein prominentes Gegenbeispiel in der Biologie ist die Abbildung von RNS-Sequenzen (Genotyp) zu sekundären RNS-Strukturen (Phänotyp). In diesem Fall ist der Genotypraum viel größer als der Raum der sekundären RNS-Strukturen [38].

Stand der Forschung

Eine frühe Arbeit zur Verwendung einer indirekten Kodierung innerhalb einer evolutionären Optimierung stammt von Kitano [21]. Hier wird eine Grammatik verwendet, um den Phänotyp aufzubauen. D.h., zur Reduktion der Dimensionalität des Genotypraumes kommt kein Lernen zum Einsatz. Kitano optimiert neuronale Netze mit Genetischen Algorithmen. Die Aufgabe der Netze ist die Modellierung eines einfachen N-M-N Kodierer/Dekodierer-Problems. Der Vergleich einer direkten Kodierung mit einer indirekten Kodierung, die mit Hilfe der vorgeschlagenen Grammatik implementiert wurde, zeigt, dass die indirekte Kodierung die Optimierung um eine Größenordnung beschleunigen kann.

Eine der ersten Arbeiten, die die Kopplung von Evolution und Lernen untersuchte, ist die Arbeit von Hinton und Nowlan [15]. In dieser wird anhand einer sehr einfach aufgebauten, aber schwer zu lösenden Aufgabe demonstriert, dass die Verwendung von Lernen innerhalb der Evolution die Form der Fitnesslandschaft für eine evolutionäre Optimierung verbessern kann. Die Aufgabe ist ein typisches „Nadel im Heuhaufen" Problem mit einer Fitnesslandschaft, die an allen Punkten, außer dem Optimum, den gleichen Wert aufweist. Eine evolutionäre Suche kann in diesem Raum keinen geeigneten Pfad zum Optimum finden und ist somit nicht besser als eine reine Zufallssuche. Die Einführung von Lernen für einen Teil der gesamten Lösungen verändert, ohne dass Informationen in die Genome zurückkodiert werden[10], die Fitnesslandschaft in der Weise, dass eine evolutionäre Suche viel schneller erfolgen kann.

Sendhoff und Kreutz [42] implementieren zusätzlich eine Entwicklungsphase – einen Wachstumsprozess – in die evolutionäre Optimierung. Darüber hinaus ist auch ein überwachter Lernprozess mit der Evolution gekoppelt. In der evolutionären Optimierung werden neuronale Netze daraufhin optimiert, eine Zeitreihenvorhersage durchzuführen. Der mit der Evolution gekoppelte Lernalgorithmus ist ein einfaches Backpropagation-Lernen. Die in diesem Lernen gefundenen Gewichte werden in das wachsende Netz zurückkodiert, in dem sie dann nur einen kleinen Teil der zu bestimmenden Gewichte ausmachen. Sendhoff und Kreutz zeigen, dass das in dieser Weise inkorporierte Lernen zusammen mit dem definierten Wachstumsprozess die Stabilität des evolutionären Optimierungsprozesses erhöht. Das wachsende Netz ist in der Lage, die zuvor gelernten Netzwerkgewichte zu nutzen, obwohl sich die Netzwerkstruktur, in die die Gewichte zurückkodiert werden, verändert hat.

Eine sehr stark neurobiologisch inspirierte Arbeit ist die von Rolls und Stringer [32]. In ihr wird die Ontogenese als ein fundamentales Konzept der Evolution neuronaler Systeme betont. In der demonstrierten evolutionären Optimierung beschränken sich Rolls und Stringer auf drei klassische Grundnetze: assoziative Netze, auto-assoziative Netze und kompetitive Netze. Die Aufgabe der Evolution war u.a. die Auswahl von verschiedenen klassischen

[10] Wie es beispielsweise bei der *Lamarckian-Evolution* getan wird.

Lernalgorithmen und die Einstellung der Anzahl der Neuronen, des Schwellwerts und der Steigung der allgemeinen Transferfunktion. Auch optimiert wurde die Verteilungsfunktion, nach der die innerhalb einer Schicht liegenden Neuronen miteinander verknüpft wurden.

Zusammenfassend kann gesagt werden, dass eine Reihe von wissenschaftlichen Arbeiten zeigen, dass eine evolutionäre Suche durch die Kopplung mit einem eingebetteten Lernprozess entscheidend verbessert werden kann. Die verwendeten Applikationen jedoch zielen in der Regel eher auf einfache Modellaufgaben. Eine anspruchsvolle 3D-Objekterkennung war bisher nicht Gegenstand einer mit Lernen gekoppelten Evolution.

Kapitel 3

Evolutionäre Optimierung des visuellen Systems

In diesem Kapitel wird die Methode der Evolutionsstrategien zum optimalen Entwurf des vorgestellten biologisch motivierten visuellen Systems eingesetzt. Zur besseren Einordnung der Schwierigkeit der visuellen Problemstellung werden zunächst Standardklassifikatoren direkt auf den Bilddaten eingesetzt. Danach wird das visuelle System nach einer Kodierung mit evolutionären Methoden auf die Objekterkennung bzw. -klassifikation optimiert (vgl. auch [34]). Nach der Darstellung der erzielten Ergebnisse wird das Konzept der Generalisierung 1. und 2. Ordnung erläutert, um die Verallgemeinerungsfähigkeit des optimierten visuellen Erkennungssystems auf andere Objektdomänen beschreiben zu können (vgl. auch [35]). In einem nächsten Schritt werden die durch den Entwurfsprozess gefundenen Nichtlinearitätsparameter und Kombinationsmerkmale analysiert. Das Kapitel schließt ab mit der Vorstellung zweier Methoden zur Steigerung der Generalisierung 2. Ordnung.

3.1 Klassifikation ohne visuelles System

Zur besseren Einordnung der Schwierigkeit der visuellen Problemstellung werden in diesem Abschnitt einige leistungsfähige Standardverfahren zur Lösung eingesetzt. Die verwendeten Klassifikatoren sind das Multi-Layer-Perceptron (MLP) und das Single-Layer-Perceptron (SLP). Als eine gute Einführung hierzu siehe [13, 4]. Als dritter Klassifikator wird der Nächste-Nachbar-Klassifikator (NNK) eingesetzt.

Multi-Layer-Perceptron

Ein universaler Funktionsapproximator mit neuronaler Struktur, der sich im Allgemeinen gut zum Einsatz in Klassifikationsaufgaben eignet, ist das Multi-Layer-Perceptron. Der Einsatz im Bereich der visuellen Klassifikation von

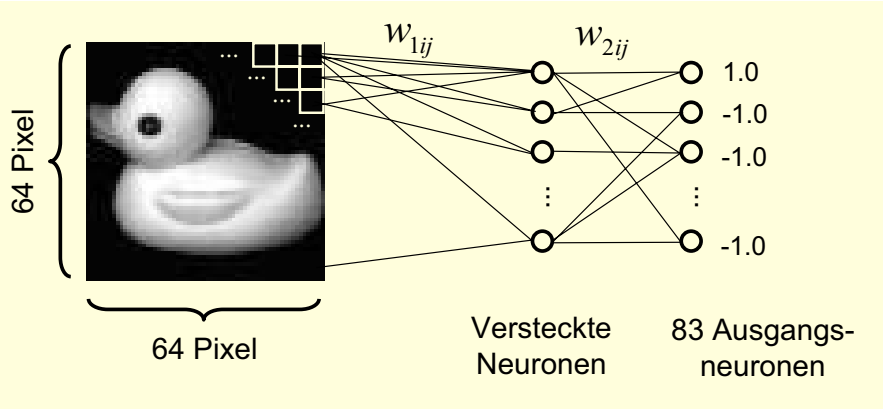

Abbildung 3.1: Schematische Darstellung des Aufbaus eines zur Objekterkennung verwendeten Multi-Layer-Perceptrons. Das Netz ist vollständig verschaltet. Die einzelnen Netzgewichte werden mit w_{1ij} und w_{2ij} bezeichnet. (Der verwendete Bias einer Schicht ist nicht dargestellt.)

Bilddaten ist jedoch problematisch. Ein Grund hierfür ist u.a. die große Dimensionalität der Bilddaten. Groß ist u.U. auch die erwünschte Anzahl der zu unterscheidenden Klassen, was wiederum zu einer großen Zahl von Ausgangsneuronen führt. Notwendigerweise beinhalten die entsprechenden MLPs eine sehr große Anzahl von zu bestimmenden Gewichtsparametern. Zwar existieren viele Trainingsalgorithmen zur Bestimmung von Netzgewichten, aber all diese Verfahren skalieren schlecht bei zunehmender Anzahl von Gewichten.

Ein anderes, größeres Problem liegt in der Generalisierungsfähigkeit der trainierten Netze. Die zu erreichende Funktion des Netzes ist nicht das „Auswendiglernen" der Trainingsdaten, sondern die Fähigkeit zur Generalisierung. Anders gesagt: es soll anhand der Trainingsdaten erkannt werden, welche Merkmale den unterschiedlichen Objekten einer Klasse gemeinsam sind. Auf diese Weise sollen dann neue Objektansichten der korrekten Klasse zugeordnet werden. Zur Demonstration dieser Problematik werden im Folgenden MLPs zur Lösung der oben (Abschnitt 2.1.2) geschilderten Klassifikationsaufgabe trainiert. Hierbei arbeiten die Netze direkt auf den Bildern der COILselect Bilddaten.

Der Aufbau eines verwendeten MLPs ist in Abbildung 3.1 schematisch dargestellt. Das Netz verfügt entsprechend der Auflösung der Inputbilder über 4096 Eingangsneuronen ($64 \times 64 = 4096$). Für die Anzahl der Neuronen der zweiten Schicht, die sogenannten *versteckten Neuronen*, wurden verschiedene Werte verglichen. Im Allgemeinen ermöglicht eine höhere Anzahl von versteckten Neuronen ein flexibleres neuronales Netz, welches damit aber auch leichter zu einer *Überanpassung (Overfitting)* der Trainingsdaten und damit zu einer schlechten Generalisierungsleistung führt. Um eine geeignete Anzahl von versteckten Neuronen zu ermitteln, wurden Netze mit 10, 20, 30, 40 und 50 versteckten Neuronen verwendet. Jedes Netz wurde 10-mal mit unterschiedlichen

Zufallsinitialisierungen trainiert. Die dritte und damit zugleich die Ausgangsschicht beinhaltet, entsprechend der Klassenanzahl, 83 Ausgangsneuronen. Die Transferfunktion der Neuronen ist der Tangens hyperbolicus. Das Netz wurde darauf trainiert, dass bei der Präsentation des ersten Objektes das erste Ausgangsneuron den Wert 1.0 annehmen soll und alle anderen Ausgangsneuronen den Wert -1.0 annehmen[1]. Bei der Präsentation des zweiten Objektes soll das zweite Ausgangsneuron den Wert 1.0 annehmen und alle anderen den Wert -1.0 usw. .

Als Trainingsverfahren wurde der sogenannte *skaliert konjugierte Gradientenabstieg* verwendet [24]. Das im Allgemeinen schneller konvergierende *Levenberg-Marquardt* Verfahren [10], das in einem folgenden Kapitel Verwendung findet, konnte hier aufgrund der enormen Größe der Netze nicht verwendet werden. Im Falle von 50 versteckten Neuronen liegt die Anzahl der zu bestimmenden Gewichte schon über 200000. Jeder der 10 Trainingsläufe wurde entweder 20000 Iterationen lang ausgeführt oder solange, bis der Betrag des Gradienten kleiner war als $1 \cdot 10^{-5}$.

Anschließend wurde jedes Netz dazu benutzt, zunächst die Trainingsbilder und dann die Testbilder zu klassifizieren. Hierbei wurde der Netzausgang so interpretiert, dass das Ausgangsneuron mit dem größten Wert bestimmt, welches Objekt erkannt wird. D.h., wenn das fünfte Ausgangsneuron den größten Wert aller Ausgangsneuronen annimmt, dann wird das präsentierte Bild als das fünfte Objekt klassifiziert.

Die Netzstruktur mit 40 versteckten Neuronen stellte sich als die leistungsfähigste heraus, sowohl was die Klassifikationsrate auf den Testbildern, als auch die Klassifikationsrate auf den Trainingsbildern anbelangt. Auf den Trainingsbildern war der mittlere Trainingsfehler 3.5% bei einer Standardabweichung von 3.9% und einem besten Fehler von 0%. Die Fehlklassifikationsrate auf den Testbildern betrug im Mittel 46.0% bei einer Standardabweichung von 6.8% und einem besten Fehler von 40.0%[2].

Diese relativ schlechte Klassifikationsrate auf den Testbildern, insbesondere im Vergleich zu dem Trainingsfehler, lässt das Problem deutlich werden, dass ein MLP zwar in der Lage ist, jede beliebige Funktion zu approximieren, dass jedoch die eigentliche Aufgabe der Generalisierung damit oft nur unzureichend gelöst ist. Eine weitere Möglichkeit – neben der Optimierung der Anzahl der versteckten Neuronen – die Generalisierungsfähigkeit eines neuronalen Netzes zu erhöhen, ist das sogenannte *Early-Stopping*. Hierbei wird während des Trainings des Netzes der Verlauf eines Test- bzw. Validierungsfehlers betrachtet. Steigt dieser signifikant an, wird das Training beendet, um ein Overfitting zu vermeiden. Eine Anwendung dieses Verfahrens im vorliegenden Fall erbrachte

[1] Diese Werte werden von der verwendeten Approximation des Tangens hyperbolicus schon für Werte von 20 bzw. -20 angenommen ($tanh(20) = 1.0$, $tanh(-20) = -1.0$).

[2] Bei einer reinen Zufallsentscheidung wäre die Wahrscheinlichkeit für eine richtige Klassifikation 1/83 und damit der Klassifikationsfehler gleich 98.8%.

jedoch auch keine Steigerung der Generalisierungsfähigkeit der MLPs.

Bezüglich der Anpassungsfähigkeit der verwendeten Netze stellt sich im Übrigen zum Teil heraus, dass eine Netzstruktur, die zwar aufgrund geringerer Flexibilität weniger gut in der Lage ist, die vorgegebenen Trainingsdaten korrekt zu klassifizieren, gleichzeitig bessere Ergebnisse (im Vergleich zu einer komplexeren Netzstruktur) in der Klassifikation der Testdaten zeigt. Solch ein weniger flexibles Netz ist beispielsweise das Single-Layer-Perceptron (SLP).

Single-Layer-Perceptron

Das verwendete Single-Layer-Perceptron ist in der gleichen Weise aufgebaut wie das zuvor diskutierte MLP mit dem Unterschied, dass die Schicht der versteckten Neuronen nicht vorhanden ist. Somit sind die Eingangsneuronen direkt über Gewichte mit den Ausgangsneuronen verbunden. Dieser Klassifikator ist auch bei optimalem Training nur in der Lage, linear separable Klassen voneinander zu trennen. Trotz dieser eingeschränkten theoretischen Leistungsfähigkeit kann das SLP über eine mit dem MLP vergleichbare Generalisierungsleistung verfügen. Bei den 10 Optimierungsläufen, die in der gleichen Weise wie das Training der MLPs zuvor ausgeführt wurden, kam es zu folgenden Ergebnissen:

Der mittlere Trainingsfehler betrug 63.0% bei einer Standardabweichung von 27.1% und einem kleinsten Fehler von 20.1%. Die Fehlklassifikationsrate auf den Testbildern betrug im Mittel 73.4% bei einer Standardabweichung von 18.6% und einem besten Fehler von 51.4%. Damit ist das SLP zwar nicht besser in der Generalisierung als das MLP mit 40 versteckten Neuronen, aber der Unterschied zwischen den mittleren Fehlern auf Trainings- und Testdaten ist kleiner. Bei dem SLP stieg der Fehler hier lediglich um 10.4% an (von 63.0% auf 73.4%), während er beim MLP von 3.5% auf 46.0% um 42.5% anstieg.

Ein Problem des SLP und des MLP ist das relativ aufwendige Training, das erforderlich ist, um die Netze zur Objekterkennung einsetzen zu können. Ein weiteres Problem, das hiermit zusammenhängt, ist die ineffiziente Art neue Objekte hinzulernen zu können. So ist es nämlich erforderlich, zur Integration eines neuen Objektes alle schon trainierten Netzgewichte erneut zu trainieren. Ein alternativer Klassifikator, der ohne eigentliches Training auskommt und der ebenso einfach neue Objektklassen integrieren kann, ist der Nächste-Nachbar-Klassifikator (NNK).

Nächster-Nachbar-Klassifikator

Eine algorithmisch sehr einfache Klassifikation basiert auf der Berechnung des nächsten Nachbarn in einem beliebigen Raum der Inputbilder. Im einfachsten, hier untersuchten, Fall wird für die notwendige Abstandsberechnung das Euklidische Abstandsmaß verwendet. Die Abstandsberechnung soll nachfolgend im Raum der rohen Bilddaten, d.h., im 4096-dimensionalen Raum der

64×64-Pixel großen Inputbilder, stattfinden. Das *Training* besteht bei dieser Form des Klassifikators lediglich darin, die Trainingsansichtsbilder als Punkte in diesem 4096-dimensionalen Raum *abzuspeichern*. Soll nun ein Testinputbild klassifiziert werden, wird der Trainingsvektor bestimmt, der den kleinsten Abstand zu dem Inputvektor hat, der also der Nächste-Nachbar (NN) ist. Dem Input wird dann die Klasse des NN zugeordnet. Der entscheidende Vorteil dieses Verfahrens ist, dass das Training keine Iterationen erfordert und deswegen praktisch keine Zeit in Anspruch nimmt. Das ist von besonderer Bedeutung für einen Einsatz innerhalb einer evolutionären Optimierungsschleife. Ein weiterer Vorteil besteht darin, dass das Objekterkennungssystem sehr einfach um eine neue Objektklasse erweitert werden kann. So bleiben die bisher gelernten Objektansichten unangetastet bestehen, und es werden lediglich die Ansichten der neuen Objektklasse in den bestehenden Raum hinzugefügt.

Das Finden des Nächsten-Nachbarn, d.h., der Klassifikationsvorgang selbst, kann in sehr hochdimensionalen Räumen mit sehr vielen Daten zwar bei der Verwendung von einfachen Algorithmen auch einige Zeit in Anspruch nehmen. Eine spürbare Entlastung liefern hier jedoch Algorithmen, die zuvor einen Suchbaum aufbauen. Zu nennen ist hier das sogenannte *KD-Tree*-Verfahren.

Der Klassifikationsfehler des Nächsten-Nachbar-Klassifikators auf der COIL-select Datenbank beträgt 31.1%. Damit ist dieser sehr einfache Klassifikator in diesem Fall besser als das beste der trainierten MLPs, dessen beste Fehlklassifikationsrate bei 40.0% liegt. Dieses Ergebnis unterstützt die in Abschnitt 2.1.2 dargelegte Herangehensweise, zur Objekterkennung einen NN-Klassifikator zu nutzen, jedoch den Raum, auf dem dieser operiert, in einer geeigneten Weise zu transformieren. Diese Transformation soll wie geschildert nach Prinzipien des menschlichen visuellen Sehsystems vorgenommen werden. Die entsprechende visuelle Merkmalshierarchie – das visuelle System – soll mit den Methoden der Evolutionären Algorithmen optimal gestaltet werden. Dazu ist es zunächst notwendig, das visuelle System in einem Genom zu kodieren.

3.2 Kodierung von Parametern und Strukturen des visuellen Systems

Um die in Abschnitt 2.1.1 beschriebene visuelle Hierarchie evolutionär optimieren zu können, ist es notwendig zu entscheiden, welche Eigenschaften der Hierarchie unverändert erhalten bleiben und welche zum Gegenstand der Optimierung werden sollen. Die zu optimierenden Eigenschaften sind in das Genom zu kodieren.

Der biologisch motivierte grundsätzliche Aufbau der visuellen Hierarchie soll bei der Optimierung in jedem Fall erhalten bleiben. D.h., die in Abschnitt 2.1.1 angegebenen Gleichungen sowie die Verwendung von orientierten Gaborfiltern bleiben unverändert. Im Fokus der Optimierung stehen hingegen

die Parametrisierung von wesentlichen Nichtlinearitäten des Systems sowie die Kombinationsmerkmale.

3.2.1 Systemnichtlinearitäten

Die Nichtlinearitäten, die wesentlich die Qualität der visuellen Verarbeitungshierarchie steuern, sind durch sechs Parameter gegeben. Diese sind:

- Die Winner-Take-Most (WTM) Parameter γ_1, γ_2, welche in die Gleichungen (2.2) und (2.7) eingehen. Sie steuern die Stärke der lateralen Kompetition von unterschiedlichen Merkmalen an identischen räumlichen Positionen in den verschiedenen Ebenen einer Schicht.

- Die Schwellwertparameter θ_1, θ_2, welche in die Gleichungen (2.3) und (2.8) eingehen. Sie steuern, ab welchem Wert einer Neuronenaktivierung diese in die folgende Schicht weiterpropagiert wird.

- Die Standardabweichungen der Gaußkerne σ_1, σ_2, welche mittelbar in die Gleichungen (2.5) und (2.9) eingehen. Sie steuern die Invarianz der Hierarchie gegenüber einer räumlichen Translation von Merkmalen.

Diese Parameter werden als Fließkommazahlen in das Genom kodiert. Die Werte sind auf die folgenden Intervalle beschränkt: $\gamma_1, \gamma_2 \in [0, 0.99]$. Hierbei bedeutet ein Wert nahe 0.99 eine sehr hohe Kompetition, d.h., dass nur die stärksten Merkmale in die nachfolgende Schicht weiterpropagiert werden. Alle schwächeren Merkmale werden an diesem Raumpunkt unterdrückt. Ein Wert nahe 0 hingegen bedeutet, dass keine Kompetition herrscht und alle Merkmale ohne Abschwächung weitergeleitet werden. Sinnvolle Werte für die Schwellwerte θ_1, θ_2 liegen in dem Intervall $[0, 3]$. Diese Grenzen ergeben sich aus der Normierung der Grauwerte der Eingangsbilder. Die möglichen Variationen der Poolingweiten sind ebenfalls auf einen sinnvollen Bereich von $[0.0001, 7]$ beschränkt. Sollte durch Mutation ein Parameter aus seinem erlaubten Bereich herausfallen, so wird stattdessen der überschrittene Grenzwert als Ausgangswert der Mutation angenommen.

3.2.2 Kombinationsmerkmale

Neben den Nichtlinearitäten der visuellen Hierarchie werden auch die Kombinationsmerkmale evolutionär optimiert. Diese sind innerhalb der Hierarchie durch die Variable $\bar{\mathbf{w}}_2^l$ (vgl. Gleichung (2.6)) repräsentiert. Hierbei wird mit $l = 1, ..., L$ jedes einzelne der L Kombinationsmerkmale indiziert. Somit ist die Dimension jedes einzelnen Kombinationsmerkmals durch die vier Ebenen der vorgelagerten C1-Schicht und durch die Größe der verwendeten räumlichen Filter $\bar{\mathbf{w}}_2^l \in \mathbb{R}^{36=4\times3\times3}$ gegeben. Jede der vier Ebenen, die mit einer der vier unterschiedlichen Orientierungen in dem Eingangsbild korrespondiert,

wird mit einem 3×3 Filter gefaltet. Im Folgenden werden zwei unterschiedliche Arten von Kombinationsfiltern betrachtet: solche ohne und solche mit negativen Werten. Im nicht negativen Fall seien die Werte des Kombinationsmerkmals folgendermaßen beschränkt: $w_{2i}^{lk} \in [0,1]$, mit $k = 1,...,4$ und $i = 1,...,36$. Im Fall der negativen Kombinationsmerkmale („neg. KM") gelte: $w_{2i}^{lk} \in [-1,1]$. Zur Optimierung werden die Werte der KM als Fließkommazahlen in das Genom kodiert. Der konzeptionelle Unterschied liegt darin, dass im Falle der negativen KM auch das Vorhandensein eines vorgelagerten Merkmals zur Abschwächung eines Kombinationsmerkmals führen kann, was im Falle von nicht negativen KM unmöglich ist.

Mit dieser Art der direkten Kodierung der Kombinationsmerkmale hat die evolutionäre Optimierung die vollständige Freiheit, die Funktionsweise aller möglichen KM auszutesten. Auf der anderen Seite resultiert diese Freiheit in den Einstellmöglichkeiten in einem sehr hochdimensionalen Parameterraum. Bei einer Anzahl von lediglich $L = 9$ KM ist die Dimension des Suchraumes (ohne die Nichtlinearitäten) 324 ($4 \cdot 3 \cdot 3 \cdot 9 = 324$). Bei $L = 50$ ist der Suchraum bereits 1800-dimensional. Mit wachsender Dimensionalität des Suchraumes wird im Allgemeinen die evolutionäre Suche immer aufwendiger[3]. Daher wird es im Allgemeinen sinnvoll sein, eine Kodierung der Kombinationsmerkmale so auszulegen, dass die Anzahl der freien Parameter möglichst klein ist. Eine Möglichkeit wird in [34] dargestellt. Eine andere wird in Kapitel 5 vorgestellt werden.

3.3 Aufbau des evolutionären Optimierungsverfahrens

Nach der Festlegung der Kodierung des visuellen Systems stellte sich in ersten Voruntersuchungen heraus, dass die Verwendung von zwei Strategieparametern $\sigma_{\mathrm{nl}}, \sigma_{\mathrm{km}}$ sowohl einer rein globalen, als auch einer individuellen Schrittweitenadaptation (was *einer* Schrittweite für jeden einzelnen Parameter entspricht; vgl. Abschnitt 2.2.2) überlegen ist. Hierbei steuert σ_{nl} die mutative Schrittweite für die Nichtlinearitäten des Systems und σ_{km} die Stärke der Mutation der Kombinationsmerkmale. Weiter zeigte sich, dass eine (μ, λ)-Evolutionsstrategie einer $(\mu + \lambda)$-Strategie vorzuziehen ist, da letztere eine stärkere Tendenz aufwies, frühzeitig in lokale Fitnessmaxima zu konvergieren. Als ein guter Kompromiss zwischen Rechenaufwand und Optimierungsgüte ergab sich für die Nachkommengröße ein Wert von $\lambda = 19$. Bei dieser Populationsgröße stellte sich ein Selektionsdruck von $\frac{19}{7} \approx 2.7$ als optimal heraus. Damit ergibt sich für die folgenden Optimierungen eine $(7, 19)$-Strategie. Au-

[3]Rechenberg zeigt, dass bei der Optimierung einer Kugelfunktion mit Hilfe einer $(1, \lambda)$-Strategie die notwendige Anzahl der Generationen zum Erreichen des Optimums linear mit der Dimension des Problems ansteigt [29].

ßerdem als sinnvoll zeigte sich für die Rekombination die folgenden jeweiligen Anwendungen: eine diskrete Rekombination für die Systemnichtlinearitäten, eine generalisierte intermediäre Rekombination für die Strategieparameter und keine Rekombination für die Kombinationsmerkmale.

Darüber hinaus ist festzulegen, über wie viele Kombinationsmerkmale L die zu optimierende Hierarchie verfügen soll. Zu beachten ist hierbei, dass mit einer steigenden Anzahl der Rechenaufwand einer Erkennung und damit die benötigte Zeit annähernd linear ansteigt. Um also Objekte möglichst schnell erkennen zu können, ist eine geringe Anzahl von KM vorteilhaft. Bei einer zu geringen Anzahl von Kombinationsmerkmalen ist jedoch der Raum, innerhalb dem die Objekte klassifiziert werden, nicht hochdimensional genug und damit die Erkennungsleistung schlecht. In den folgenden Untersuchungen werden daher drei unterschiedliche Werte für L verwendet: $L = 9, 36, 50$. Der Fall mit $L = 50$ Kombinationsmerkmalen ermöglicht einen direkten Vergleich mit der Leistungsfähigkeit des visuellen Systems nach Optimierung der Nichtlinearitäten mit Hilfe einer einfachen Rastersuche und einer Bestimmung der Kombinationsmerkmale anhand einer einfachen Heuristik [49]. Alle folgenden evolutionären Optimierungsläufe wurden 10-mal mit unterschiedlichen Anfangsinitialisierungen für eine Dauer von 400 Generationen durchgeführt.

3.4 Ergebnisse der Optimierung

Die Ergebnisse der evolutionären Optimierung für visuelle Hierarchien mit einer unterschiedlich großen Anzahl von Kombinationsmerkmalen sind in Tabelle 3.1 dargestellt. Mit „pos. KM" sind die auf das Intervall $[0, 1]$ beschränkten Kombinationsmerkmale bezeichnet und mit „neg. KM" die auf das Intervall $[-1, 1]$ beschränkten. Aufgeführt sind die Fehlklassifikationsraten auf der COIL20 Datenbank, jeweils mit dem besten und dem mittleren Ergebnis, sowie der Standardabweichung der jeweils 10 Optimierungsläufe.

Es wird ein minimaler Fehler von 6.5% (bei $L = 9$ negativen Kombinationsmerkmalen) erreicht. Der Fehler des Ausgangssystems („AGS") von Wersing und Körner [49] konnte damit von 14.4% um 7.9% gesenkt werden. Das entspricht einer Verbesserung um über 50%. Das Ausgangssystem wurde in den Nichtlinearitäten mit Hilfe einer einfachen Rastersuche und bezüglich der Kombinationsmerkmale mit einem heuristischen Aufzählungsverfahren optimiert (für nähere Angaben s. [49]). Außerdem wurden bei dem Ausgangssystem $L = 50$ Kombinationsmerkmale verwendet und bei dem besten der optimierten Systeme lediglich neun. Damit ist die Erkennungsgeschwindigkeit bei dem verbesserten visuellen System ca. doppelt so hoch (ca. 50 ms auf einem Pentium III 850MHz).

Ein Vergleich der beiden unterschiedlich beschränkten Kombinationsmerkmale zeigt, dass eine Einbeziehung von negativen Werten („neg. KM") die mitt-

		Fehler COIL20 [%]		
	L	b	m	s
AGS	50	14.4	–	–
pos. KM	9	7.9	8.6	0.5
	36	7.7	8.3	0.5
	50	7.3	8.6	0.8
neg. KM	9	**6.5**	8.1	1.2
	36	7.1	**7.8**	0.5
	50	7.1	8.1	0.7

Tabelle 3.1: Ergebnisse der evolutionären Optimierungen. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. AGS=Ausgangssystem.

lere Fehlklassifikationsrate um 0.5% senkt. Diese Verbesserung des Systems gilt konstant für jede der drei unterschiedlich großen KM-Mengen ($L = 9, 36, 50$). Statistisch signifikant[4] ist die Verbesserung allerdings nur für $L = 36$. Allerdings ist der Trend, dass negative Kombinationsmerkmale von Vorteil sind, auch klar bei $L = 9$ und $L = 50$ zu erkennen. Diese leichte Verbesserung des Ergebnisses kann damit begründet werden, dass die erweiterten Möglichkeiten der Zusammenstellung von Kombinationsmerkmalen für die visuelle Erkennungshierarchie von Vorteil ist. Dieser Vorteil überkompensiert auch eine etwaige Erschwernis der evolutionären Suche in einem vergrößerten Parameterraum.

Bezüglich der unterschiedlichen Anzahl von verwendeten Kombinationsmerkmalen stellt sich heraus, dass die beste mittlere Erkennungsrate in beiden Ansätzen bei $L = 36$ liegt (wobei die Unterschiede relativ gering sind). Eine Anzahl von 36 Kombinationsmerkmalen scheint ein guter Kompromiss zwischen einer zu kleinen Adaptionsfreiheit des Systems und einem zu großen Suchraum für die evolutionäre Optimierung zu sein.

3.5 Generalisierung 1. und 2. Ordnung

Die Aufgabe eines technischen Systems zur Objekterkennung ist im Allgemeinen die Folgende: Das System soll anhand von einer kleinen Menge von Beispiel- oder Trainingsansichten verschiedener Objekte eine Verknüpfung von Bildansicht zu Objektidentität erlernen. Entscheidend hierbei ist, dass nach erfolgtem Lernen nicht nur die Trainingsansichten richtig klassifiziert werden können, sondern auch nie gesehene Testansichten. Dazu ist sicherzustellen, dass der Erkenner nicht nur die Trainingsdaten „auswendig lernt" sondern viel-

[4] Als Test für die statistische Signifikanz wird stets der Student-t-Test verwendet. Wenn nicht anders angegeben, wird ein Signifikanzlevel von 95% angenommen.

mehr die zugrunde liegende Abbildung zwischen Ansichten und Objektidentität erlernt. Damit das System gut von den Trainingsdaten auf neu eingehende Testdaten bzw. -ansichten generalisieren kann, sollte die Auswahl der Trainingsdaten möglichst repräsentativ sein. Neben der „Unvollständigkeit" der Trainingsdaten, d.h. der Tatsache, dass diese nur eine kleine Stichprobe aller möglichen Daten darstellen, stellt das Auftreten von Rauschen auf den Trainingsdaten ein Problem dar.

Um lernen zu können, muss das System über eine anpassungsfähige Struktur verfügen. Mit zunehmender Flexibilität dieser Strukturen können immer komplexere Zusammenhänge modelliert und immer genauer die Trainingsdaten erlernt werden. Auf der anderen Seite steigt mit immer größerer Flexibilität auch die Anfälligkeit des Systems, nur die Trainingsdaten gut zu erkennen, aber eine schlechte Generalisierung gegenüber den Testdaten aufzuweisen. Das System hat sich zu stark an die Trainingsdaten angepasst, man spricht in diesem Zusammenhang von einer *Überanpassung* oder von einem *Overfitting*. Dieses prinzipielle Problem, dass ein wenig anpassungsfähiges System schlecht einen Zusammenhang erlernen kann und ein zu anpassungsfähiges System überangepasst wird, wird als das sogenannte *Bias-Variance-Dilemma* [9] bezeichnet.

Bei der Suche nach einer ausgewogenen Flexibilität der lernenden Struktur wird daher oftmals das Verfahren der *Kreuzvalidierung* oder *Cross-Validation* verwendet. Hierbei wird die strukturelle Flexibilität des Systems beispielsweise so lange erhöht (und damit der Modellierungsfehler auf den Trainingsdaten beständig gesenkt), bis der Fehler auf einer Menge von Test- oder Validierungsdaten ansteigt. In diesem Moment kann man nämlich davon ausgehen, dass es zu einem Overfitting der Trainingsdaten gekommen ist. Dasselbe Verfahren kann auch zum vorzeitigen Abbruch des Lernverfahrens verwendet werden. In diesem Fall spricht man auch von einem *Early-Stopping*.

Bei der Erkennung der Objekte mit Hilfe des visuellen Systems besteht das **Lernen** aus dem **Abspeichern** der Trainingsansichten in dem hochdimensionalen Merkmalsraum, der durch die visuelle Hierarchie aufgespannt wird. Um eine Testansicht zu erkennen, wird diese ebenfalls in diesen Merkmalsraum abgebildet. Dort wird mit Hilfe des Euklidischen Abstands die nächste Trainingsansicht gesucht. Diese bestimmt, als welches Objekt die Testansicht erkannt wird (vgl. Abbildung 2.3 Seite 24). Durch die evolutionäre Optimierung der Nichtlinearitäten und Kombinationsmerkmale kommt es zu einem weiteren Lernvorgang. Will man jetzt also die Güte des visuellen Systems – bzw. der entworfenen Kombinationsmerkmale und Nichtlinearitäten – in Bezug auf die Generalisierungsleistung beurteilen, so muss auch dieses evolutionäre Lernen mit in die Überlegungen einbezogen werden.

Dies könnte dadurch geschehen, dass alle Ansichten, die innerhalb der beiden Lernvorgänge:

1. Abspeichern der prototypischen Ansichten im C2-Raum

2. Evolutionärer Entwurfsprozess der Nichtlinearitäten und Kombinationsmerkmale

verwendet werden, als Trainingsdaten angesehen werden und ein weiterer Satz von Objektansichten zur Validierung eingesetzt wird. Anhand der Erkennungsleistung auf den Validierungsdaten könnte dann die Fähigkeit des Systems zur Generalisierung abgeschätzt werden. Viel mehr interessiert jedoch die Frage, wie gut das optimierte visuelle System auch zur Erkennung von ganz anderen Objekten eingesetzt werden kann.

Aus diesem Grund wird im Folgenden das Konzept der Generalisierung 1. und 2. Ordnung[5] zur Evaluierung des optimierten visuellen Systems eingeführt. Die Vorgehensweise bei der evolutionären Optimierung des visuellen Systems ist die folgende: Das System wird in einem Genom kodiert und innerhalb einer evolutionären Schleife optimiert. Die Fitness eines Individuums ergibt sich aus der Fähigkeit des visuellen Systems, bisher noch nicht gesehene Ansichten eines Objektes zu erkennen. Zu dieser Evaluation muss jedes Individuum in ein visuelles System dekodiert, mit Trainingsansichten belehrt und anschließend mit Testansichten getestet werden. Je besser die Klassifikationsleistung des Netzes ist, desto höher ist seine Fitness. Die Klassifikationsleistung ist ein Maß dafür, wie gut das System über unterschiedliche Ansichten eines Objektes generalisieren kann. Durch die Verwendung der Generalisierungsleistung als Fitness werden die Testansichten mit in die Evolutionsschleife einbezogen. Die evolutionäre Optimierung hat damit die Möglichkeit, das visuelle System und die darin befindlichen Merkmale an diese Testansichten anzupassen. Wie stark diese Möglichkeit gegeben ist, hängt in großem Maße von der Adaptierbarkeit des Systems ab. Grundsätzlich jedoch ist diese Möglichkeit gegeben und deshalb wird diese Klassifikationsleistung im Folgenden *Generalisierung 1. Ordnung* genannt. Diese Generalisierung gibt an, wie gut das visuelle System nach dem Lernen von Trainingsansichten auf Testansichten generalisieren kann. Hierbei stammen alle Objekte aus der Datenbank, die bei dem evolutionären Entwurf verwendet wurde. Damit ist das System besonders auf das Lernen und Erkennen dieser Objekte angelegt. Um jedoch die Möglichkeit zu haben, das System mit Hilfe des Lernens um neue Objekte erweitern zu können, ohne die Struktur erneut anpassen zu müssen, sollte das System auch über eine gute *Generalisierung 2. Ordnung* verfügen. Diese gibt an, wie gut das visuelle System allgemein in der Lage ist, Objekte von beliebigen Datenbanken anhand von Trainingsansichten zu lernen und anhand von Testansichten wiederzuerkennen. Die allgemeine bilddomänenübergreifende Funktionsweise beruht zum einen auf einer robusten Einstellung der Systemnichtlinearitäten und zum anderen auf einer robusten und wiederverwendbaren Auslegung der

[5]Die in dieser Arbeit benutzte Unterscheidung von Generalisierung 1. und 2. Ordnung ist ähnlich der, die von Hüsken et al. [19] vorgeschlagen wurde. Weiter ist zu bemerken, dass die gebräuchlicheren Begriffe Test- und Validierungsfehler ungeeignet sind, da man sich hierbei auf Untermengen einer Datenbank bezieht, was in dieser Arbeit nicht der Fall ist.

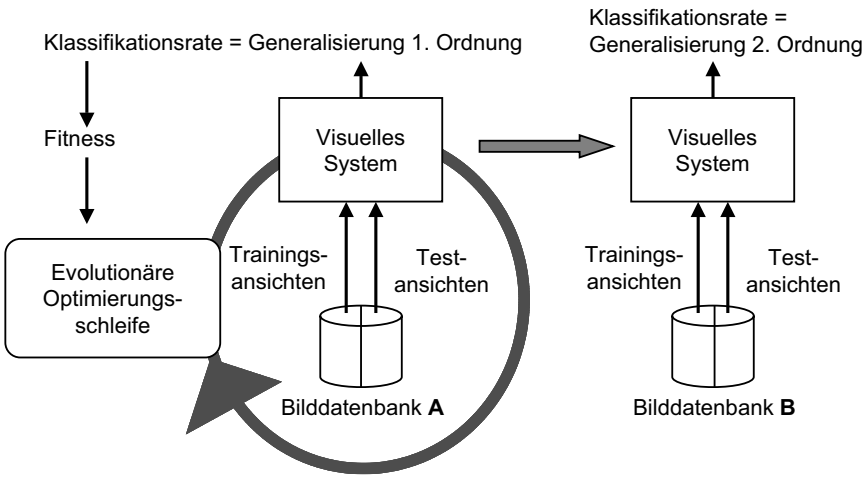

Abbildung 3.2: Konzept der Generalisierung 1. und 2. Ordnung. Zur Ermittlung, wie bilddomänenübergreifend die Erkennungsleistung des optimierten visuellen Systems ist, wird das System nach der evolutionären Optimierung der Nichtlinearitäten und Kombinationsmerkmale auf einer anderen Bilddatenbank B getestet.

Kombinationsmerkmale. Zur Abschätzung dieser Generalisierung 2. Ordnung wird das – mit Hilfe einer Datenbank A – evolutionäre optimierte visuelle System mit Trainingsansichten von Objekten einer anderen Datenbank B trainiert und mit Testansichten getestet. Die Klassifikationsleistung, die das System auf dieser Datenbank B erreicht, gibt Aufschluss über die Generalisierungsleistung 2. Ordnung des visuellen Systems. Das Konzept der Generalisierung 1. und 2. Ordnung ist schematisch in Abbildung 3.2 dargestellt.

Im folgenden Abschnitt soll nun überprüft werden, wie gut die Generalisierungsleistung 2. Ordnung der entworfenen visuellen Systeme ist. Dabei ist zu beachten, dass der bisherige evolutionäre Entwurfsprozess in keiner Weise Einfluss auf die Verbesserung dieser Generalisierung 2. Ordnung genommen hat.

3.6 Ergebnisse der Generalisierung 2. Ordnung

Zur Bestimmung der Generalisierung 2. Ordnung wird zunächst die COILselect Datenbank verwendet. Im Gegensatz zu der COIL20 Datenbank, die bei der evolutionären Optimierung des visuellen Systems verwendet wurde, besteht diese nicht aus 20, sondern aus 83 Objekten. Die Objekte sind andere als bei der COIL20, jedoch relativ ähnliche. Dadurch wird die Wahrscheinlichkeit erhöht, dass die gefundenen Systemnichtlinearitäten und auch die Kombinationsmerkmale noch mit einer relativ hohen Performanz einsetzbar sind. In Tabelle

		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
AGS	50	14.4	–	–	28.5	–	–
pos. KM	9	7.9	8.6	0.5	24.2	27.6	3.0
	36	7.7	8.3	0.5	23.2	26.5	2.2
	50	7.3	8.6	0.8	23.2	24.8	1.5
neg. KM	9	**6.5**	8.1	1.2	22.8	26.3	2.4
	36	7.1	**7.8**	0.5	22.9	**24.2**	1.8
	50	7.1	8.1	0.7	**22.4**	24.3	1.7

Tabelle 3.2: Ergebnisse der evolutionären Optimierungen. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. AGS=Ausgangssystem.

3.2 sind nun neben den bisherigen Fehlklassifikationsraten auf der COIL20 Datenbank (vgl. Tabelle 3.1) die Fehler auf der COILselect Datenbank dargestellt. Dies geschieht wie zuvor jeweils mit dem besten und dem mittleren Ergebnis sowie der Standardabweichung der jeweils 10 Optimierungsläufe. Die COIL20 Datenbank wurde innerhalb der Evolutionsschleife benutzt, und damit ist die hier erreichte Erkennungsleistung gleich der Generalisierung 1. Ordnung. Die Erkennungsleistung des fertig optimierten Systems auf der COILselect Datenbank entspricht der Generalisierung 2. Ordnung (vgl. Abschnitt 3.5).

Die beste Generalisierung 2. Ordnung liegt bei einem Fehler von 22.4%. Verglichen mit dem Ausgangssystem („AGS") von Wersing und Körner [49] stellt das eine Absenkung des Fehlers von 28.5% um 6.1% dar. Damit ist die Absenkung des Fehlers nicht mehr ganz so groß wie bei der COIL20 Datenbank (7.9%), aber die Verbesserung des Systems liegt immer noch über 20%.

Ein Vergleich der beiden unterschiedlich beschränkten Kombinationsmerkmale zeigt, dass eine Einbeziehung von negativen Werten („neg. KM") wie schon zuvor bei der Generalisierung 1. Ordnung die mittlere Fehlklassifikationsrate verbessert. Der klar in mittlerem und bestem Wert erkennbare Trend ist wie zuvor nur bei $L = 36$ statistisch signifikant. Bezüglich der unterschiedlichen Anzahl von verwendeten Kombinationsmerkmalen stellt sich wie bei der Generalisierung 1. Ordnung heraus, dass die beste mittlere Erkennungsrate bei $L = 36$ liegt (wobei auch hier die Unterschiede gering sind).

Betrachtet man die Flexibilität bzw. die Anpassungsmöglichkeit des visuellen Systems, so wird diese durch die Anzahl L der Kombinationsmerkmale variiert. Eine Erhöhung von L entspricht auch einer Erhöhung der Anpassungsmöglichkeiten des Systems. Im Falle einer zu hohen Flexibilität des Systems kommt es im Allgemeinen zu einem Overfitting der Trainingsdaten und damit zu einem Anstieg des Fehlers auf den Validierungsdaten. In dem vorliegenden

Abbildung 3.3: Beispielbilder der durch die AT&T Research Labs in Cambridge (UK) bereitgestellten ORL-Gesichtsdatenbank.

Fall arbeitet das betrachtete evolutionäre „Lernen" noch eine Stufe über dem Lernen des visuellen Systems. An der Stelle eines Validierungsfehlers wird der Generalisierungsfehler des Systems auf einer völlig anderen Datenbank (Generalisierung 2. Ordnung) betrachtet. Dieser Fehler verhält sich im Mittel wie der Generalisierungsfehler 1. Ordnung und zeigt damit, dass die Flexibilität des Systems bei $L = 36$ nicht zu einem „Overfitting 2. Ordnung" führt (denn sonst hätte der Generalisierungsfehler 2. Ordnung ansteigen müssen). Und auch bei $L = 50$ kann noch nicht von einem wirklichen Anstieg gesprochen werden.

Die Objekte der COILselect Datenbank ähneln in ihrem Aussehen denen der COIL20. Es stellt sich die Frage: Wie gut ist die Klassifikationsleistung auf einer Datenbank, die einer völlig anderen Objektdomäne angehört? Eine solche Datenbank ist beispielsweise die ORL-Gesichtsdatenbank, bereitgestellt von den AT&T Research Labs in Cambridge (UK). Diese Datenbank enthält Bilder von 40 unterschiedlichen Gesichtern. Jedes Gesicht wurde 10-mal in unterschiedlichen Haltungen und Gesichtsausdrücken aufgenommen. Die Grauwertbilder, die im Original in einer Auflösung von 92×112 Pixel vorliegen, wurden für die Erkennung auf die zuvor benutzte Größe von 64×64 Pixel umskaliert. Ansonsten wurden weder Änderungen an den Bildern aus der Datenbank noch bei den Parametereinstellungen des visuellen Systems vorgenommen. Die Klassifikationsaufgabe wurde für die zuvor auf die COIL20 optimierten Systeme so vorgenommen, dass immer je 2 Ansichten (immer das 1. und das 6. Bild) zum Training verwendet wurden und die restlichen acht Ansichten zu erkennen waren. Abbildung 3.3 zeigt je sechs der 10 verschiedenen Bilder von vier der 40 unterschiedlichen Gesichter.

Die Fehlklassifikationsraten sind (analog zu Tabelle 3.2) in Tabelle 3.3 aufgetragen. An erster Stelle ist zu bemerken, dass die auf der ORL-Datenbank erzielten Ergebnisse verglichen mit Lawrence [23] gut sind (vgl. Legende von

		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler ORL [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
pos. KM	9	7.9	8.6	0.5	19.1	27.4	7.4
	36	7.7	8.3	0.5	16.6	25.0	6.0
	50	7.3	8.6	0.8	**13.8**	21.0	5.4
neg. KM	9	**6.5**	8.1	1.2	18.2	24.3	5.7
	36	7.1	**7.8**	0.5	16.3	21.3	7.5
	50	7.1	8.1	0.7	14.7	**19.6**	5.0

Tabelle 3.3: Ergebnisse der evolutionären Optimierungen. Zum Test der Generalisierungsfähigkeit 2. Ordnung wurde hier die ORL-Gesichtsdatenbank verwendet. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. Im Vergleich dazu erreicht Lawrence et al. [23] eine Fehlklassifikationsrate von 17.0% auf der ORL-Datenbank.

Tabelle 3.3), insbesondere wenn man beachtet, dass die Optimierung des Erkennungssystems auf einer völlig anderen Bilddomäne stattgefunden hat.

Weiter ist festzustellen, dass die Ergebnisse ähnlich denen auf der COILselect Datenbank sind. Auch hier zeigen die negativen Kombinationsmerkmale eine im Mittel bessere Performanz. Auch hier ist eine Steigerung der Leistung mit steigendem L zu verzeichnen. Anders jedoch als bei der COILselect ist hier bei $L = 36$ neg. KM keine Sättigung zu beobachten. Zu bemerken ist auch, dass die Ergebnisse bei der ORL-Datenbank eine viel größere Streuung aufweisen als die Ergebnisse auf der COILselect Datenbank. Das lässt sich damit erklären, dass die Übertragbarkeit der Ergebnisse von COIL20 zu ORL weniger robust ist als die von COIL20 zu COILselect, da es sich hierbei auch um sehr unterschiedliche Bilddomänen handelt. Es zeigt sich außerdem (nicht in der Tabelle dargestellt), dass in der Mehrzahl der Fälle die Systeme, die schon auf der COILselect eine gute Generalisierung 2. Ordnung bewiesen haben, auch auf der ORL Datenbank eine gute Generalisierungsleistung zeigen.

Zusammenfassend lässt sich bemerken, dass obwohl bei dem evolutionären Entwurf nur eine bestimmte Datenbank verwendet wurde, die erhaltenen visuellen Systeme gut auch auf anderen Bilddatenbanken arbeiten können. Diese Leistung kann mit der biologisch motivierten robusten Grundstruktur des visuellen Systems erklärt werden.

Overfitting

Zum Abschluss der Untersuchungen zum Generalisierungsverhalten der optimierten visuellen Systeme soll noch mit einer Standardmethode untersucht werden, ob es bei dem evolutionären Lernen zu einem Overfitting gekom-

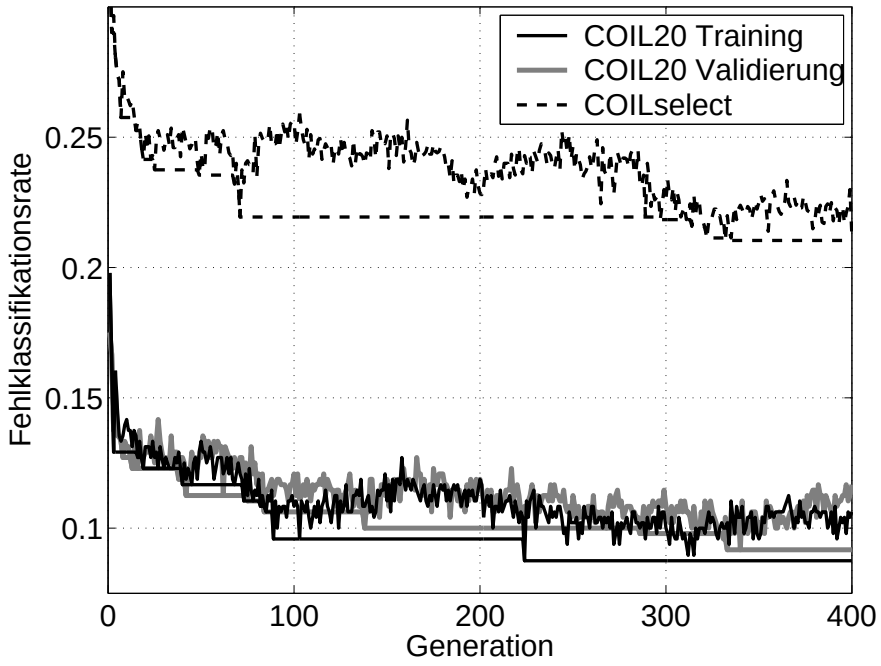

Abbildung 3.4: Fehlklassifikationsraten des besten Individuums bei einem repräsentativen Optimierungslauf ($L = 9$ neg. KM). Dargestellt ist der Fehler auf der Trainingsmenge, der Validierungsmenge (beide aus COIL20) und der Fehler auf der COILselect Datenbank. Neben dem besten Individuum ist auch der jeweils kleinste erreichte Fehler aufgetragen.

men ist. Hierzu wird eine Menge von 24 Objektansichten als Validierungsmenge bestimmt. Diese Ansichten werden weder als Prototypen des visuellen Systems noch zur Fitnessbestimmung innerhalb der evolutionären Optimierung verwendet. Die Rotationswinkel dieser Validierungsansichten sind $\alpha_{\text{Validierung}} = 10° + i \cdot 15°$ mit $i = 0, 1, 2, ..., 23$.

Im Falle eines Overfittings würde während der evolutionären Optimierung der Fehler auf den Trainingsdaten des evolutionären Lernens ($\alpha_{\text{Training}} = 5° + i \cdot 15°$ mit $i = 0, 1, 2, ..., 23$) beständig abnehmen, während der Fehler auf den Validierungsdaten beständig ansteigen würde. Dieses Verhalten konnte in keinem der durchgeführtem Optimierungsläufe beobachtet werden, vielmehr sinkt der Validierungsfehler mehr oder weniger analog zu dem Testfehler ab. Analog dazu wurde auch der Fehler auf der COILselect Datenbank im Zeitverlauf beobachtet. Ein Ansteigen dieses Fehlers könnte als ein Overfitting 2. Ordnung bezeichnet werden. Allerdings ergibt sich auch hier ein vergleichbares Bild: Der Fehler auf der COILselect sinkt vergleichbar mit dem Validierungsfehler ab. Ein typischer Verlauf für $L = 9$ negative Kombinationsmerkmale ist in Abbildung 3.4 dargestellt.

In dieser Arbeit wird die Flexibilität oder Anpassungsmöglichkeit des ler-

nenden visuellen Systems nur in geringem Umfang verändert, und zwar durch eine unterschiedliche Anzahl L von verwendeten Kombinationsmerkmalen. Aber auch für das flexibelste visuelle System bei $L = 50$ ist der Verlauf der Fehler vergleichbar mit dem aus Abbildung 3.4. Auch hier ist somit weder ein Overfitting 1. noch eines 2. Ordnung zu erkennen. D.h., obwohl die Erkennungsleistung auf einer speziellen Datenbank innerhalb des evolutionären Lernens kontinuierlich verbessert wird, kommt es nicht zu einer Verschlechterung des visuellen Systems bezüglich der Erkennungsleistung auf einer anderen Datenbank. Dieses günstige Verhalten des Systemlernens kann damit begründet werden, dass zwar viele Parameter des visuellen Systems gelernt werden können, jedoch die Grundstruktur starr genug ist, um eine Überanpassung zu vermeiden.

3.7 Analyse der optimierten Systeme

Eine andere wichtige Frage, die sich nach der Durchführung der verschiedenen Optimierungsläufen stellt, ist die nach der Beschaffenheit der optimierten visuellen Systeme. Wurden immer ähnliche Nichtlinearitäten und Kombinationsmerkmale gefunden? Hierzu sollen im Folgenden zunächst die Nichtlinearitäten betrachtet werden. In Abbildung 3.5 sind die Nichtlinearitäten von 10 optimierten visuellen Hierarchien dargestellt. Mit Linien verbunden ist immer ein gefundener Satz von Nichtlinearitäten. Jeder Satz entstammt dem besten Individuum, das innerhalb eines vollständigen Optimierungslaufes gefunden wurde. Betrachtet werden die 10 Optimierungen mit $L = 9$ negativen Kombinationsmerkmalen, also das Szenario, innerhalb dessen das System mit der besten Erkennungsrate auf der COIL20 Datenbank gefunden wurde. Das hier dargestellte Ergebnis ist weitgehend repräsentativ, verglichen mit den restlichen Ergebnissen. Es zeigt sich, dass die einzelnen gefundenen Nichtlinearitäten innerhalb von relativ großen Intervallen verteilt sind, und dass keine der Nichtlinearitäten immer auf einen kleinen Wertebereich beschränkt bleibt. Weiter kann festgestellt werden, dass die Weite des ersten Poolings, gesteuert über σ_1, im Mittel doppelt so groß ist wie die des zweiten Poolings, gesteuert über σ_2. Beachtet man, dass das zweite Pooling auf Bildausschnitten arbeitet, die auf ein Viertel der Größe des originalen Bildausschnittes, auf denen das erste Pooling arbeitet, herunterskaliert sind, so hat man es – bezogen auf das Inputbild – nicht mit einer Halbierung, sondern mit einer Verdopplung des Poolings zu tun.

Weiter kann beobachtet werden, dass sich bei der Einstellung des WTM-Parameters γ_1 und des Schwellwertes θ_1 zwei unterschiedliche „Strategien" herauszubilden scheinen. Alle Systeme müssen einen gewissen Grad an Selektivität bezüglich der unterschiedlichen Richtungsmerkmale aufbauen. Die erste Strategie, die von den besten vier Individuen der 10 Optimierungsläufe gewählt

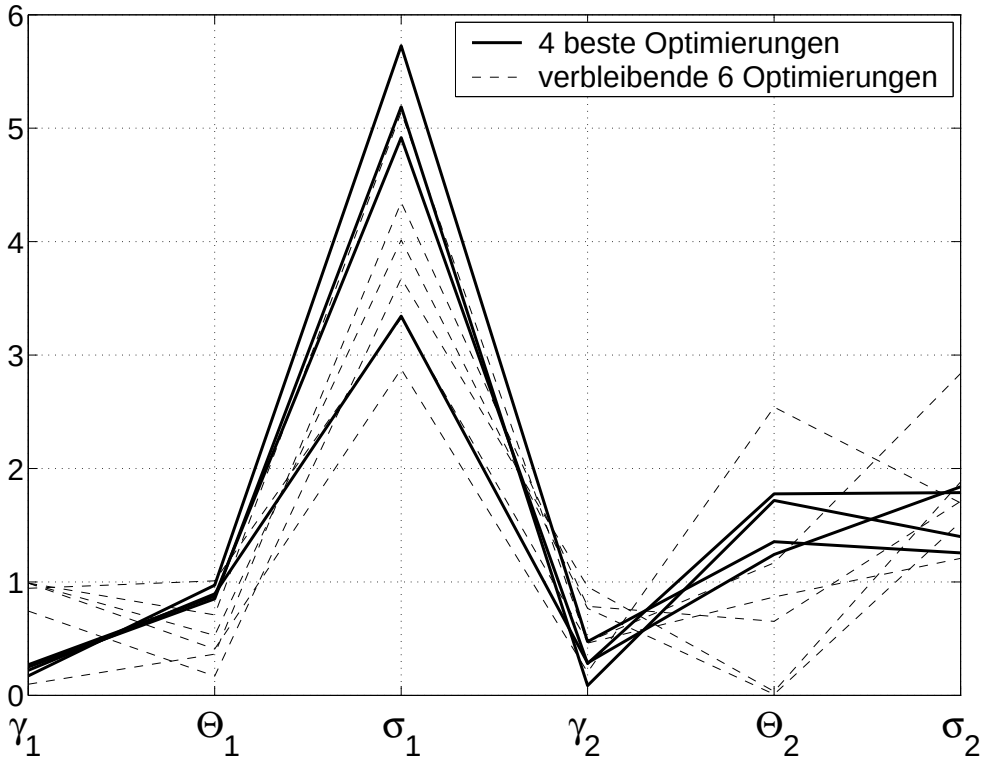

Abbildung 3.5: Werte der Nichtlinearitäten des jeweils besten Individuums eines der 10 Optimierungsläufe bei $L = 9$ negativen KM. Die sechs Nichtlinearitäten, die jeweils zu einem visuellen System gehören, sind mit Linien verbunden.

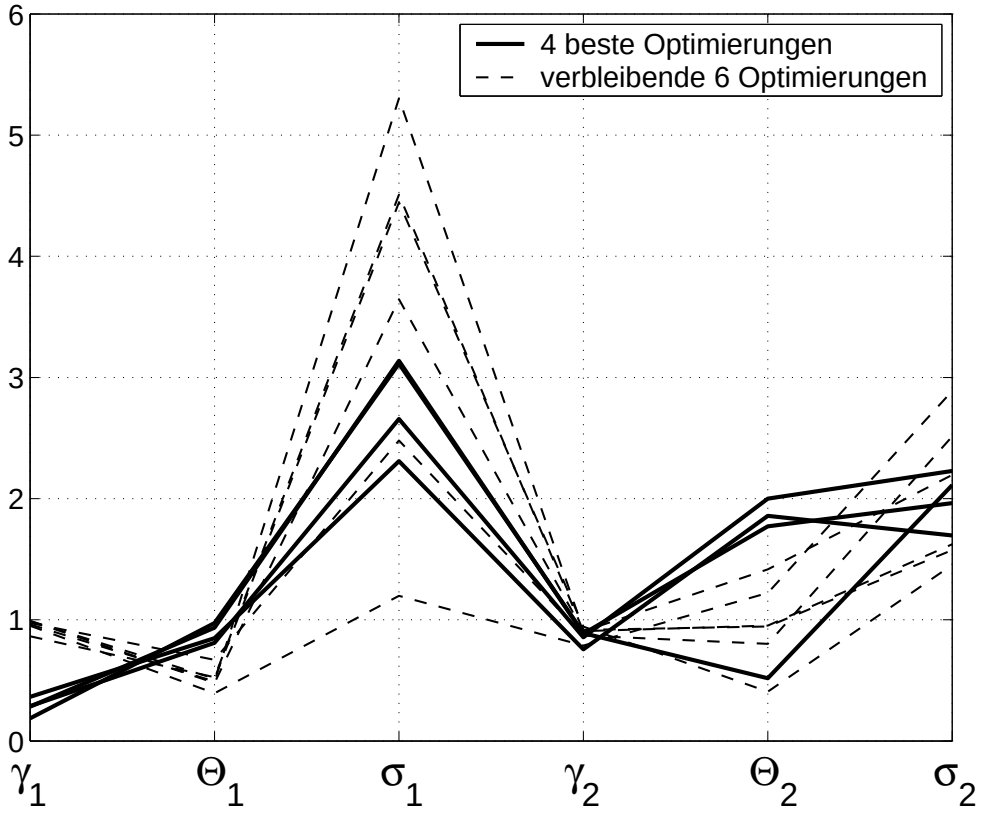

Abbildung 3.6: Werte der Nichtlinearitäten des jeweils besten Individuums eines der 10 Optimierungsläufe bei $L = 50$ positiven KM. Die sechs Nichtlinearitäten, die jeweils zu einem visuellen System gehören, sind mit Linien verbunden.

wurde, beinhaltet eine zuerst geringe Selektivität, implementiert durch einen niedrigen Wettbewerbswert γ_1, gefolgt von einer hohen Selektivität, implementiert durch einen hohen Schwellwert θ_1. Die zweite Strategie, die zumindest von vier der restlichen sechs Individuen verfolgt wurde, implementiert die Selektivität genau auf die umgekehrte Weise. D.h. eine hohe Selektivität, bewirkt durch eine starke Kompetition, wird gefolgt von einer eher geringen Selektivität, bewirkt durch einen kleinen Schwellwert. Zu bemerken ist, dass das leicht bessere Abschneiden der ersten Strategie auch in den anderen Optimierungsergebnissen zu beobachten ist.

In Abbildung 3.6 ist dieselbe Untersuchung für das Ergebnis mit $L = 50$ positiven Kombinationsmerkmalen dargestellt. Auch hier ist eine starke Variation der Nichtlinearitäten zu verzeichnen. Gemeinsam ist beiden Ergebnissen die Herausbildung der zwei Selektivitätsstrategien. Diese ist in dem Fall ($L = 50$ pos. KM) sogar noch deutlicher zu erkennen. Ebenso ist zu erkennen, dass auch hier die Stärke des Poolings sich ähnlich verhält wie im Falle von $L = 9$ neg. KM.

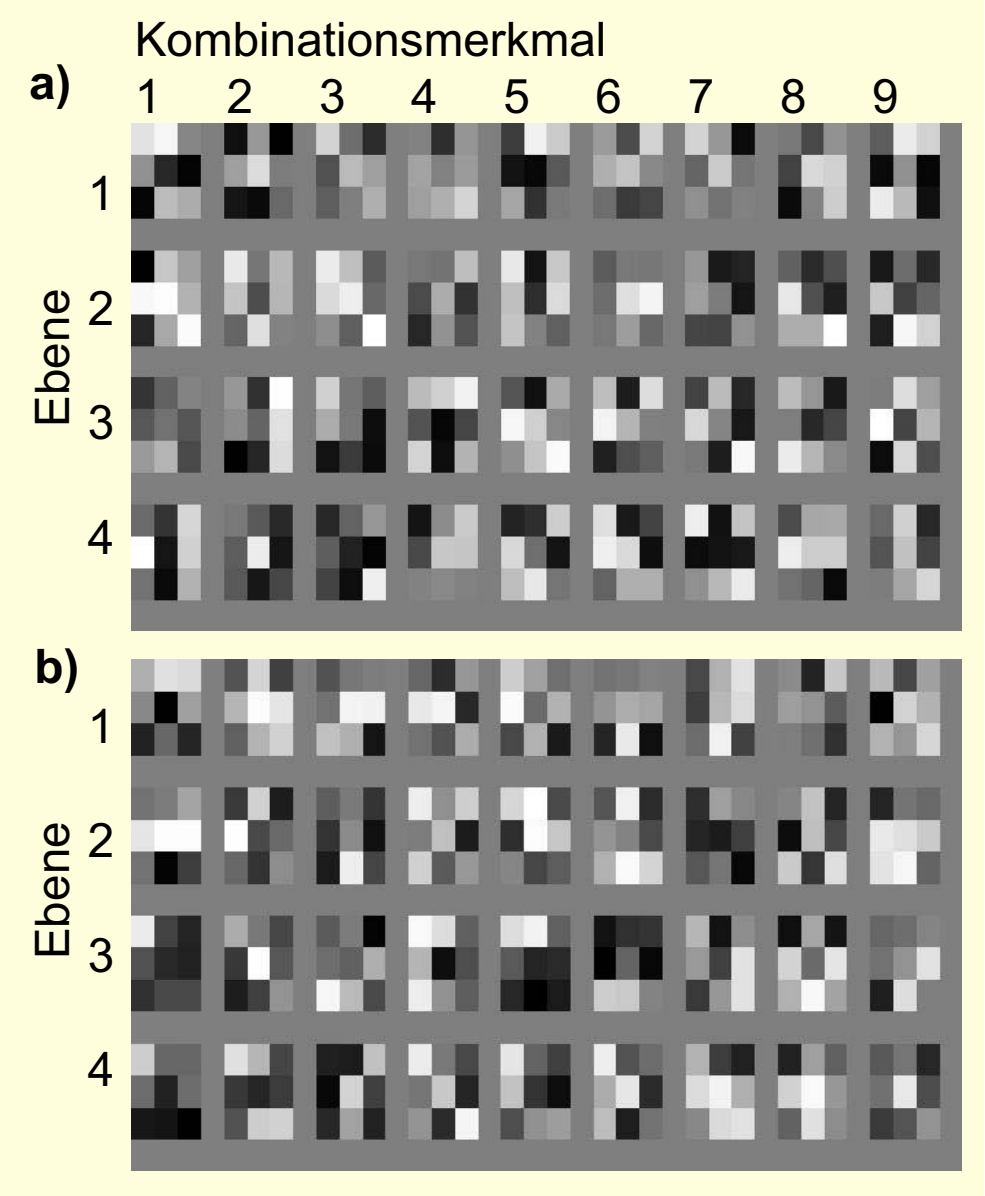

Abbildung 3.7: Optimierte Kombinationsmerkmalsbänke für den Fall $L = 9$ negative Merkmale. Die Werte reichen von -1 (schwarz) bis +1 (weiß). In den Spalten stehen die unterschiedlichen Merkmale, in den Zeilen sind die jeweiligen Ebenen der Merkmale dargestellt. a) bester Optimierungslauf b) zweitbester Optimierungslauf.

Der Vergleich der gefundenen Kombinationsmerkmale ist schwieriger als der der Nichtlinearitäten. In Abbildung 3.7 sind die zwei Kombinationsmerkmalsbänke für den erst- und zweitbesten[6] Lauf ($L = 9$ neg. KM) dargestellt. Um einen Vergleich zwischen zwei Mengen von Kombinationsmerkmalen oder kurz Merkmalsbänken, durchführen zu können, wird im Folgenden das Abstandsmaß D_{KM} definiert. Dieses Maß ist invariant unter der Permutation von einzelnen Merkmalen innerhalb einer Bank, da die Reihenfolge auch keine Auswirkung auf die Erkennungsleistung des visuellen Systems hat. Das Distanzmaß $D_{KM}(B1, B2)$ zwischen zwei unterschiedlichen Bänken B1 und B2 mit einer gleichen Anzahl L von Merkmalen sei wie folgt definiert: Beginnend mit dem ersten Merkmal der Bank $B1$ wird das im Euklidischen Abstand nächste Merkmal der Bank $B2$ gesucht. Der entsprechende Abstand $d_1^{KM}(B1, B2)$ zwischen diesen beiden Merkmalen wird gemessen (vgl. Abbildung 3.8). Anschließend werden beide Merkmale aus den entsprechenden Merkmalsbänken entfernt. Dieser Prozess wird solange wiederholt, bis alle Merkmale aus den Bänken entfernt worden sind. Um das Maß symmetrisch zu halten, werden anschließend die beiden Bänke vertauscht und der ganze Prozess erneut durchgeführt. Dabei werden insgesamt $2L$ Abstände berechnet. Zur Berechnung von $D_{KM}(B1, B2)$ summiert man diese Abstände auf und teilt das Ergebnis durch zwei. Das Maß ist damit definiert durch:

$$D_{KM}(B1, B2) = \frac{1}{2}\left[\sum_{i=1}^{L} d_i^{KM}(B1, B2) + \sum_{i=1}^{L} d_i^{KM}(B2, B1)\right] . \qquad (3.1)$$

Hierbei definiert $d_i^{KM}(B1, B2)$ den Euklidischen Abstand des i-ten Merkmals der Merkmalsbank $B1$ zum im Euklidischen Abstand nächsten Merkmal der Bank $B2$. Das resultierende Maß $D_{KM}(B1, B2)$ ist gleich Null für Merkmalsbänke, die über einen identischen Satz von Merkmalen, unabhängig von der Reihenfolge, verfügen. Je unterschiedlicher die Merkmale sind, desto größer ist der Wert des Distanzmaßes.

Mit Hilfe dieser Distanz ist es möglich, die Ähnlichkeit von Kombinationsmerkmalsbänken zu quantifizieren. Dies soll im Folgenden exemplarisch für die jeweils beste Bank (Endergebnis je eines der 10 Optimierungsläufe) für den Fall $L = 9$ negative Kombinationsmerkmale erfolgen. Um die Distanzwerte einordnen zu können, werden zunächst 10 Merkmalsbänke mit gleichverteilten Zufallszahlen erzeugt. Dann wird für jede mögliche Paarung das Distanzmaß ermittelt. Es ergibt sich die mittlere Distanz $\bar{D}_{KM} = 40.3$, bei einem geringsten Abstand von $D_{KM}^{min} = 37.8$ und einer Standardabweichung von $D_{KM}^{std} = 0.9$. Um ein besseres Verständnis zu erhalten, wie sensitiv sich der Abstand zu der Klassifikationsleistung verhält, wurden zu der Kombinationsmerkmalsbank der beiden besten Optimierungsläufe ($L = 9$ negative KM)

[6] Wenn nicht anders bemerkt, bezieht sich die Angabe „bester Lauf" auf die Generalisierung 1. Ordnung.

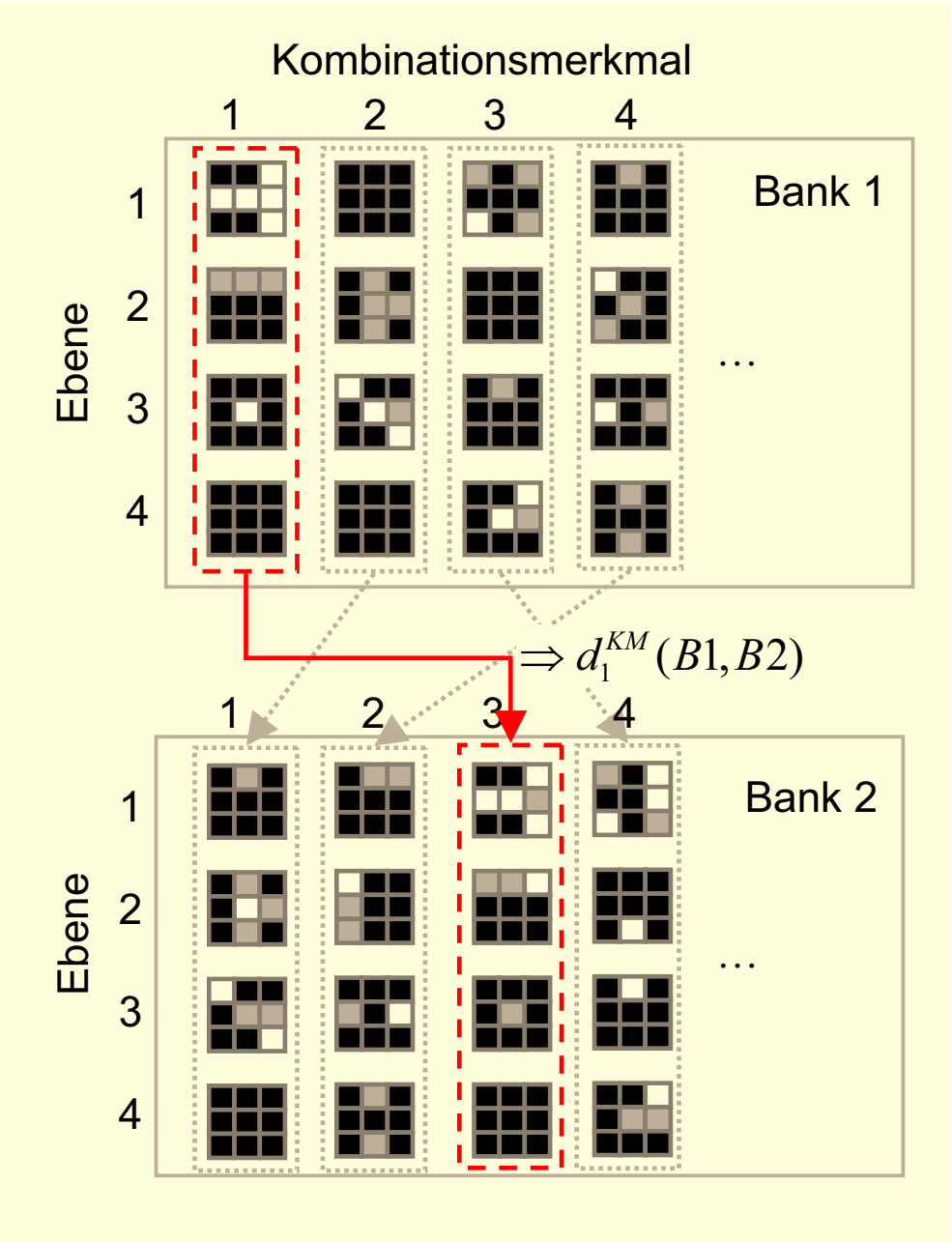

Abbildung 3.8: Schematische Darstellung der Ermittlung des Abstandsmaßes zwischen zwei Kombinationsmerkmalsbänken. Beginnend mit dem ersten Kombinationsmerkmal (KM) der Bank 1 wird das im Euklidischen Abstand nächste KM der Bank 2 gesucht (im dargestellten Fall das KM 3). Der Abstand zwischen diesen beiden Merkmalen ist $d_i^{KM}(B1, B2)$. Anschließend werden beide Merkmale aus den Bänken entfernt und der Abstand zwischen dem zweiten KM der Bank 1 und dem nächsten KM der Bank 2 bestimmt (im dargestellten Fall das KM 1). Auch hier wird der Abstand gemessen. Der Prozess wird solange fortgesetzt, bis kein Merkmal mehr übrig ist. Anschließend werden beide Bänke vertauscht und der Prozess wird noch einmal wiederholt.

L=9 neg. KM		KM+N(0,0.025)			KM+N(0,0.05)			KM+N(0,0.1)			zufällige KM		
		b	m	s	b	m	s	b	m	s	b	m	s
Bester Lauf (6.5 %)	D	1.3	1.4	0.07	2.4	2.6	0.1	4.9	5.2	0.2	40.5	41.2	0.6
	Fehler COIL20 [%]	6.9	7.7	0.5	7.3	8.1	0.6	8.1	9.7	0.8	11.9	13.4	1.2
Zweitbester Lauf (6.5 %)	D	1.3	1.4	0.07	2.4	2.6	0.1	4.8	5.2	0.2	39.1	40.0	0.6
	Fehler COIL20 [%]	7.9	8.3	0.3	7.5	8.4	0.5	8.3	8.7	0.4	12.5	14.8	1.9

Tabelle 3.4: Distanzen („D") und Klassifikationsfehler der beiden besten visuellen Systeme ($L = 9$ negative Kombinationsmerkmale) nach dem Verrauschen der Kombinationsmerkmale („KM") durch Addition normalverteilter Zufallszahlen unterschiedlicher Standardabweichung. Die Fehlklassifikationsraten der unverrauschten Systeme sind jeweils 6.5% (aufgeführt in Klammern). Es wurden je 10 Experimente durchgeführt, wobei „m" den mittleren, „s" die Standardabweichung und „b" den besten bzw. kleinsten Wert angibt.

normalverteilte Zufallszahlen addiert. Die verwendeten Standardabweichungen der Zufallszahlen sind: $\sigma_{noise} = 0.025, 0.05, 0.1$. Dies wurde 10-mal für die drei verschiedenen Rauschstärken durchgeführt. Zuletzt wurde die Bank vollständig durch eine Zufallsbank ersetzt. Nach dem Stören der ursprünglichen Bank mit Rauschen wurde der Abstand zur Ursprungsbank und die neue Fehlklassifikationsrate des veränderten visuellen Systems berechnet. Das Ergebnis dieser Untersuchung ist in Tabelle 3.4 zu finden.

Es zeigt sich, dass der Klassifikationsfehler, der in beiden optimierten Erkennungssystemen bei 6.5% liegt, schon durch die Zugabe von relativ geringem Rauschen stark ansteigt. Daraus kann geschlussfolgert werden, dass es sich bei den gefundenen Optima um relativ enge Bereiche in der Fitnesslandschaft handelt.

Die anschließende Analyse der Abstände der 10 optimierten Merkmalsbänke ergibt einen mittleren Abstand (nicht in Tabelle 3.4 dargestellt) von $\bar{D}_{KM} = 40.1$ und einen minimalen Abstand von $D_{KM}^{min} = 38.1$ bei einer Standardabweichung von $D_{KM}^{std} = 1.0$. Vergleicht man diese Werte mit den Werten, die sich bei den Zufallsbänken ergeben haben, so kann festgestellt werden, dass sich die 10 Kombinationsmerkmalsbänke, die jeweils dem besten visuellen System einer der 10 Optimierungsläufe angehören, annähernd so stark voneinander unterscheiden wie Bänke mit Zufallszahlen. Damit weisen Bänke, die eine ähnliche Klassifikationsleistung erzielen, eine hohe Variationsbreite auf. Oder anders ausgedrückt: Es wurde bei jedem Optimierungslauf eine andere optimale Bank gefunden, die sehr wenig Ähnlichkeit mit einer bereits gefundenen aufweist. Der Fitnessraum der Kombinationsmerkmalsbänke scheint daher mit in etwa gleich guten Optima bestückt zu sein, die weit im Raum verteilt sind. Eine mögliche zweidimensionale Fitnesslandschaft, die diesen Ei-

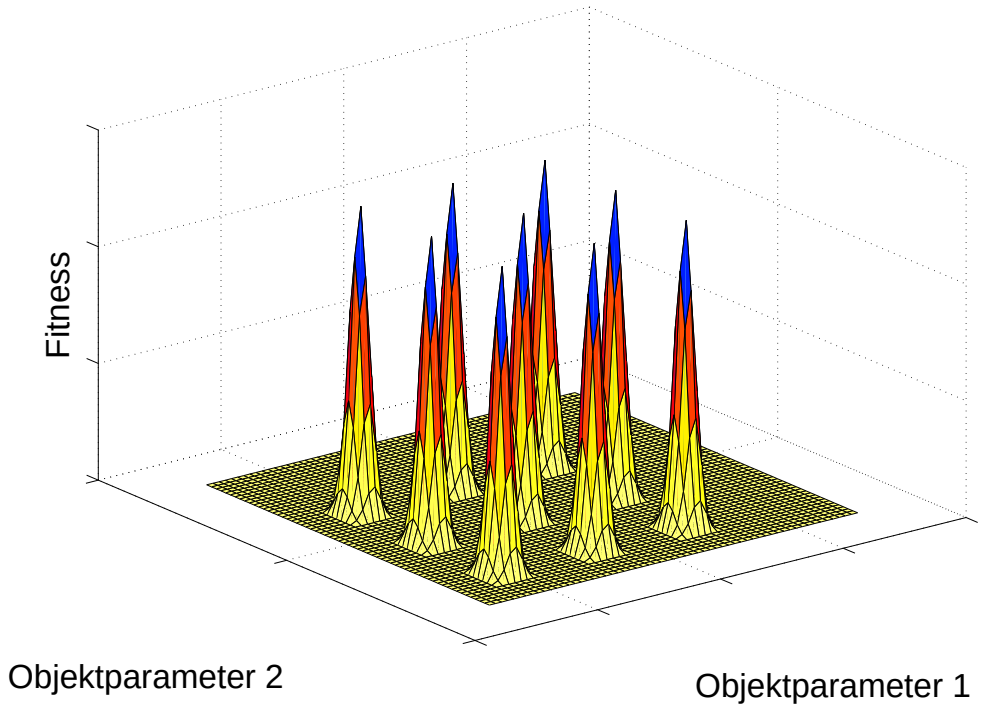

Abbildung 3.9: Eine mögliche zweidimensionale Fitnesslandschaft mit den bei den Untersuchungen gefundenen Eigenschaften.

genschaften entspricht, könnte wie in Abbildung 3.9 dargestellt aussehen. Das ist allerdings nur eine von verschiedenen Möglichkeiten. Eine andere wäre die Anordnung der Optima auf einem Ring.

Man könnte sich vorstellen, dass ein anderes Abstandsmaß existiert, das imstande ist, die gesamte Komplexität des visuellen Systems abzubilden. Dieses Maß würde dann nicht nur wie das vorliegende Maß (vgl. Gleichung (3.1)) für kleine Abstände eine ähnliche Erkennungsleistung garantieren, sondern auch für ähnliche Erkennungsleistungen ähnliche Abstände.

Zusammenfassend kann gesagt werden, dass es – innerhalb des erlaubten Variationsraumes – sehr viele unterschiedlich aufgebaute visuelle System gibt, die alle eine gute Funktionsweise aufweisen. Dieses robuste Verhalten kann als eine direkte Folge des biologisch inspirierten Aufbaus des Systems verstanden werden. Es ist wahrscheinlich, dass diese strukturelle Robustheit eine wichtige Eigenschaft in dem evolutionären Entwurfsprozess der Natur ist.

Im Folgenden wird der Zeitverlauf der besten Optimierung (mit $L = 9$ neg. KM) betrachtet. Hierzu werden in Abbildung 3.10 die Fehlklassifikationsrate, die Strategieparameter und die Nichtlinearitäten während der gesamten Optimierung dargestellt. Gleichzeitig gezeigt werden die sieben besten Individuen einer Generation. Auf diese Weise wird auch die Variationsbreite der Elternpopulation, die ja auf sieben festgelegt war, veranschaulicht. Typisch ist die festzustellende Abnahme der Mutationsschrittweite der Nichtlinearitäten σ_{nl}.

Weiter kann man auch erkennen, dass nach einer anfänglichen sehr kurzen Adaptionsphase (beachte hierzu die großen Parameterschwankungen bei der Darstellung der Nichtlinearitäten in der Nähe von 0) sich die meisten Nichtlinearitäten insgesamt nur noch wenig verändern. Die Fehlklassifikationsrate ist innerhalb der ersten 50 Generationen stark abgefallen. Bei der weiteren Optimierung sind jetzt nur noch kleinere Erfolge zu erzielen. Zur Feineinstellung der Nichtlinearitäten regelt die Evolutionsstrategie daher σ_{nl} weiter herunter. Weitere Verbesserungen des Systems können jetzt aus der Optimierung der Kombinationsbänke erzielt werden.

Vergleichend hierzu sei der beste Lauf des Szenarios mit $L = 50$ positiven Kombinationsmerkmalen herangezogen (vgl. Abbildung 3.11). Auch in diesem Fall kann man erkennen, dass die Fehlklassifikationsrate innerhalb der ersten 50 Generationen schnell absinkt. Ebenso fällt die Mutationsschrittweite σ_{nl} kontinuierlich ab. In diesem Fall ist zudem auch noch ein stärkerer Abfall der Mutationsschrittweite σ_{km} zu erkennen. Insbesondere nachdem sich γ_1 nach ca. 200 Generationen auf einen kleinen Wert adaptiert hat und auch die Adaption der anderen Nichtlinearitäten weitgehend abgeschlossen ist, scheint ein Feintuning der Kombinationsmerkmale mit Hilfe einer kleinen Mutationsschrittweite sinnvoll zu sein.

3.8 Verbesserung der Generalisierung 2. Ordnung

Vergleicht man die Ergebnisse der Generalisierung 1. und 2. Ordnung miteinander (z.B. in Tabelle 3.2 Seite 57) so ist ein deutlicher Leistungsabfall zu bemerken. Dieser rührt in diesem speziellen Fall zum einen daher, dass zur Überprüfung der Generalisierung 2. Ordnung die COILselect Datenbank mit 83 Objekten genommen wurde und damit die Erkennungsrate bei gleich komplizierten Objekten schon schlechter sein muss als bei der COIL20 Datenbank mit 20 Objekten. Jedoch ist auch prinzipiell zu erwarten, dass auch bei gleicher Anzahl und Schwierigkeit der Objekte die Generalisierung 2. Ordnung schlechter ausfallen wird, da die evolutionäre Optimierung die Möglichkeit hatte, das Erkennungssystem an die Objekte der Datenbank A (bisher COIL20) anzupassen.

Hier stellt sich die Frage, wie stark die evolutionäre Optimierung in der Lage ist, spezielle Objektinformationen in das visuelle System einzuarbeiten, bzw. wie stark der Grad der Spezialisierung bezüglich der Datenbank A ist. Die Anpassungsmöglichkeiten des visuellen Systems hierzu, nämlich über die Nichtlinearitäten und die schon sehr abstrakten Kombinationsmerkmale scheinen gering zu sein. Zum einen daher, weil die Anzahl dieser freien Parameter sehr klein gegenüber der Anzahl der Netzwerkelemente ist. Zum anderen ist das Anpassungsvermögen der Kombinationsmerkmale durch die kleine Ausdehnung der Einzelfilter auf 3×3 Pixel, begrenzt. Zur Klärung der Frage des

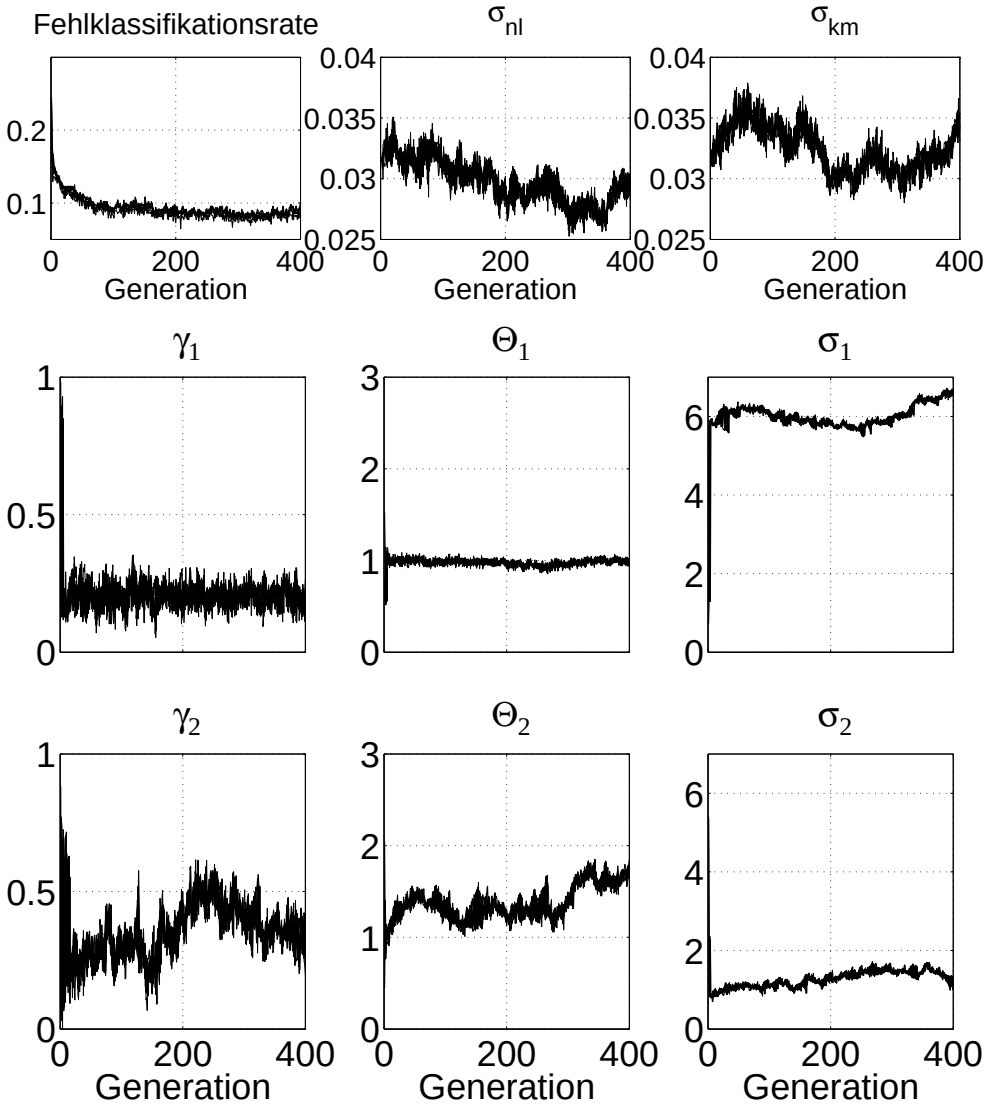

Abbildung 3.10: Zeitverlauf von Fehlklassifikationsrate (negative Fitness), Strategieparametern $\sigma_{\mathrm{nl}}, \sigma_{\mathrm{km}}$ und den Nichtlinearitäten der 7 besten Individuen während des besten Optimierungslaufes mit $L = 9$ und negativen KM. Die 7 besten Individuen, die in der nächsten Generation zu Eltern werden, sind übereinander dargestellt und visualisieren auf diese Weise die Variationsbreite der Elternpopulation im Parameterraum.

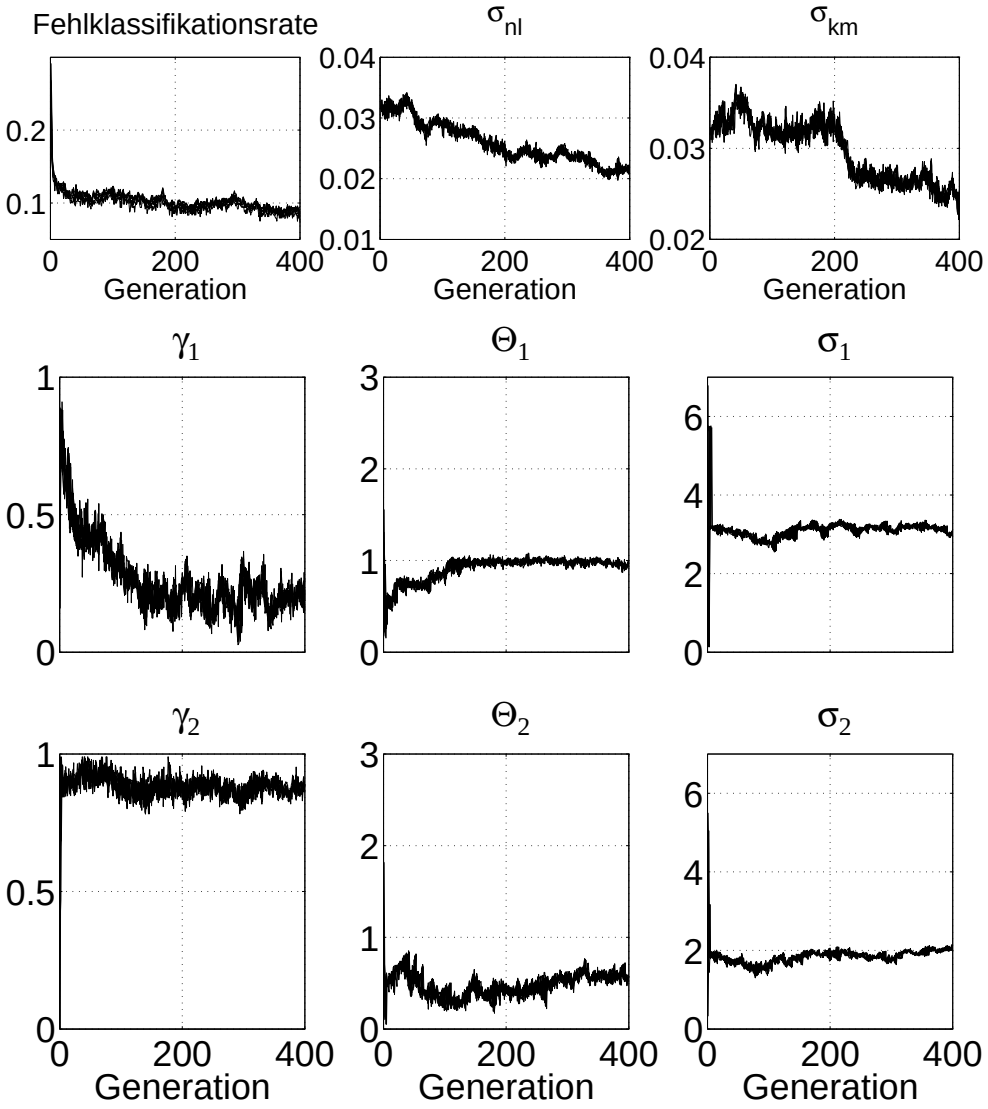

Abbildung 3.11: Zeitverlauf von Fehlklassifikationsrate (negative Fitness), Strategieparametern $\sigma_{\mathrm{nl}}, \sigma_{\mathrm{km}}$ und den Nichtlinearitäten der 7 besten Individuen während des besten Optimierungslaufes mit $L = 50$ und positiven KM. Die 7 besten Individuen, die in der nächsten Generation zu Eltern werden, sind übereinander dargestellt und visualisieren auf diese Weise die Variationsbreite der Elternpopulation im Parameterraum.

		Fehler COILselect [%] (Generalisierung 1. Ordnung)		
	L	b	m	s
AGS	50	28.5	–	–
neg.	9	17.6 (22.8)	19.1 (26.3)	0.7 (2.4)
KM	50	**16.6** (22.4)	**18.0** (24.3)	0.6 (1.7)
		Fehler COIL20 [%] (Generalisierung 2. Ordnung)		
AGS	50	14.4	–	–
neg.	9	**11.5** (6.5)	**13.8** (8.1)	1.1 (1.2)
KM	50	12.5 (7.1)	**13.8** (8.1)	1.0 (0.7)

Tabelle 3.5: Ergebnisse der Optimierung unter Verwendung der direkten Kodierung. COILselect wurde im Gegensatz zu bisher als Datenbank A (Generalisierung 1. Ordnung) und COIL20 wurde als Datenbank B (Generalisierung 2. Ordnung) eingesetzt. Zum einfacheren Vergleich sind in Klammern die Ergebnisse aus Tabelle 3.2 eingetragen, die bei dem vertauschten Gebrauch der Datenbanken erhalten wurden. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. AGS=Ausgangssystem [49].

Anpassungsvermögens werden die beiden Objektdatenbanken im Folgenden vertauscht. D.h., während der evolutionären Optimierung wird als Datenbank A nun die COILselect Datenbank verwendet und zum Test der Generalisierungsfähigkeit 2. Ordnung wird nun als Datenbank B die COIL20 verwendet.

Die Einstellungen für die Optimierung seien die gleichen wie in Kapitel 3 für $L = 9$ und $L = 50$ mit negativen Kombinationsmerkmalen. Auch hier werden 10 unabhängige Optimierungsläufe mit zufälliger Initialisierung durchgeführt. Die Ergebnisse sind in Tabelle 3.5 eingetragen.

Zum einfacheren Vergleich sind die Ergebnisse, die bei der vertauschten Verwendung der Datenbanken erzeugt wurden (vgl. Tabelle 3.2), in Klammern aufgeführt. Das beste visuelle System erreicht – trotz der geringen Anpassungsmöglichkeit des Systems – einen Fehler von 16.6% auf der 83 Objekte umfassenden COILselect Datenbank. Dieser Wert stellt ein gutes Ergebnis dar, insbesondere wenn man es mit der Fehlklassifikationsrate des Ausgangssystems („AGS") von 28.5% vergleicht. Aber auch andere leistungsfähige gegenwärtige Erkennungssysteme erreichen vergleichbare Fehler auf der nur wenig erweiterten COIL100 Datenbank (Ein Vergleich mit diesen Systemen wird in Kapitel 5.5 folgen.).

Bezüglich des datenbankspezifischen Anpassungsvermögens des visuellen Systems kann man feststellen, dass im Falle von $L = 9$ die durchschnittliche Performanz auf COILselect um 7.2% besser wird und zugleich um durchschnitt-

lich 5.7% auf COIL20 abnimmt. Gewinn und Verlust halten sich damit in etwa die Waage. Damit ist der Performanzanteil, der auf eine spezifische Anpassung an die jeweiligen Objekte der Datenbank A zurückzuführen ist, ca. 6.5%. Das ist ein relativ hoher Wert, wenn man die starken Beschränkungen betrachtet, unter denen sich das visuelle System anpassen kann. So bieten die Nichtlinearitäten kaum eine Möglichkeit, sich an die einzelnen Objekte einer Datenbank anzupassen. Die frei parametrisierten Kombinationsmerkmale bieten zwar größere Anpassungsmöglichkeiten; diese sind aber, durch die kleine Ausdehnung der Einzelfilter (3×3) und das verwendete Weight-Sharing, begrenzt.

Im Falle von $L = 50$ gewinnt man auf COILselect 6.3% und verliert wieder 5.7% auf COIL20. Damit hat man auch hier einen Wert von ca. 6% für die spezifische Anpassung. Die etwas höhere Performanz von 18% im Vergleich zu 19.1% bei $L = 9$ scheint plausibel, da das visuelle System zur freien Einstellung nun über 50 Kombinationsmerkmale verfügt. Die Tatsache, dass bei COIL20 als Datenbank A keine höhere Performanz bei 50 Merkmalen zu beobachten ist, kann damit begründet werden, dass für 20 Objekte u.U. nicht mehr als 9 Kombinationsmerkmale sinnvoll sind. Somit kann eine evolutionäre Optimierung auch bei einem visuellen System mit 50 Kombinationsmerkmalen keine weitere Verbesserung erzielen.

Die Frage, die jedoch in jedem Fall bleibt, ist: Wie kann die evolutionäre Optimierung weiter in der Art verändert werden, dass das optimierte visuelle System eine noch robustere Erkennung aufweist? Eine weitere Erhöhung der Robustheit des Systems könnte sich dann auch in einer verbesserten Generalisierung 2. Ordnung niederschlagen. Eine Möglichkeit könnte darin bestehen, schon bei der Optimierung der Erkennungsleistung auf Datenbank A Systeme, die eine robustere Erkennung aufweisen, mit einer höheren Fitness zu belohnen.

3.8.1 Regularisierung durch Verallgemeinerung der Fitnessfunktion

Die bei dem visuellen System zum Einsatz kommende Objekterkennung basiert im letzten Schritt auf der Ermittlung des Abstands zum nächsten Nachbarn. Besteht der kleinste Abstand zu einem gelernten Repräsentanten des gleichen Objektes, so wird die neue Objektansicht korrekt klassifiziert. Besteht jedoch der kleinste Abstand zu einem gelernten Repräsentanten eines anderen Objektes, dann kommt es zu einer Fehlklassifikation. Eine Form größerer Robustheit wäre es nun beispielsweise, wenn der Abstand zu dem nächsten falschen Objektrepräsentanten möglichst groß wäre. Diese Form der robusten Erkennung ist in Abbildung 3.12 dargestellt.

Um diese Form der Robustheit in dem visuellen Systemen zu erhöhen, muss diese Eigenschaft mit in die Fitnessevaluation eingebunden werden. Aus Teil c) der Abbildung 3.12 wird auch deutlich, dass es nicht ausreicht, im Falle einer korrekten Erkennung eine große Distanz zur nächsten Ansicht eines

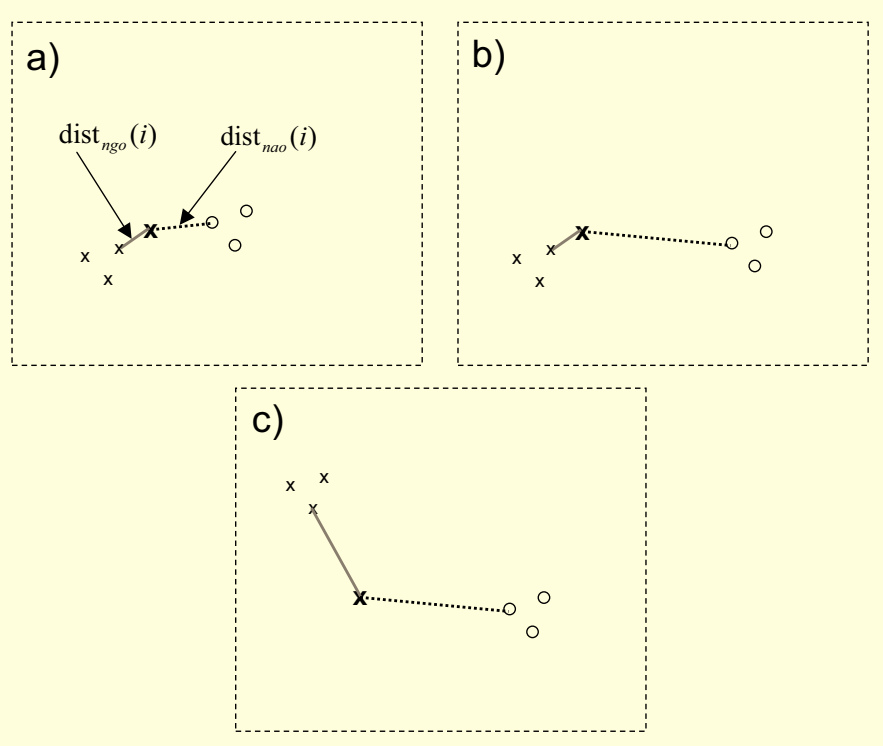

Abbildung 3.12: Schematische Darstellung einer korrekten Objekterkennung aufbauend auf einer Nächsten-Nachbar-Klassifikation in einem Merkmalsraum. Der Abstand der zu erkennenden Objektansicht i zur nächsten schon gelernten Objektansicht des gleichen Objektes wird mit $\text{dist}_{ngo}(i)$ bezeichnet (dargestellt durch eine durchgezogene Linie). Der Abstand zur nächsten Objektansicht einer schon gelernten Objektansicht eines anderen Objektes wird mit $\text{dist}_{nao}(i)$ bezeichnet (dargestellt durch eine gestrichelte Linie). a) „normale" Erkennung einer unbekannten Objektansicht (großes „X") b) robuste Erkennung einer unbekannten Objektansicht c) „normale" Erkennung bei großen Werten von $\text{dist}_{ngo}(i)$ und $\text{dist}_{nao}(i)$.

anderen Objektes dist_{nao} zu haben. Denn in diesem Falle würde die robuste Erkennung in b) gleich der weniger robusten in c) bewertet. Vielmehr ist entscheidend, dass die Differenz und somit die „Sicherheitsmarge" der beiden Abstände möglichst groß bei der korrekten Erkennung und möglichst klein bei der fehlerhaften Erkennung ist. Hierzu sei die Fitness nicht mehr lediglich die negative Fehlklassifikationsrate. Die bisherige Fitness war definiert durch:

$$\text{Fitness} = -\frac{1}{n}\sum_{i=1}^{n}(\text{Anzahl der Fehlklassifikationen}) \; . \tag{3.2}$$

Hierbei ist n gleich der Anzahl der zu klassifizierenden Objektansichten. Bei der Optimierung handelt es sich um ein Maximierungsproblem. Der maximal zu erreichende Wert ist 0.0. Das gilt auch weiterhin bei der neuen Fitnessfunktion, die jedoch zusätzlich die Robustheit der Erkennung mit einbezieht. Diese ist definiert durch:

$$\text{Fitness} = \begin{cases} -\frac{1}{2n}\sum_{i=1}^{n} 1 - (\text{dist}_{nao}(i) - \text{dist}_{ngo}(i))^q \\ \text{für } \text{dist}_{nao}(i) > \text{dist}_{ngo}(i) \\ -\frac{1}{2n}\sum_{i=1}^{n} 1 + (\text{dist}_{ngo}(i) - \text{dist}_{nao}(i))^q \\ \text{für } \text{dist}_{nao}(i) < \text{dist}_{ngo}(i) \end{cases} \; . \tag{3.3}$$

Mit $\text{dist}_{ngo}(i) \in [0,1]$ ist der kleinste Abstand der zu klassifizierenden Ansicht i zur nächsten schon gelernten Objektansicht desselben Objektes bezeichnet (ngo=nächstes gleiches Objekt). $\text{dist}_{nao}(i) \in [0,1]$ bezeichnet den kleinsten Abstand zur nächsten schon gelernten Objektansicht eines anderen Objektes (nao=nächstes anderes Objekt).

Der Wert $q \in [0,1]$ bestimmt die Gewichtung der Fitnessfunktion bezüglich der beiden Ziele: 1. hohe Klassifikationsrate und 2. robuste Erkennung (=ein möglichst großer Abstand vom nächsten anderen Objekt). Für $q = 0$ erhält man:

$$\text{Fitness} = \begin{cases} -\frac{1}{2n}\sum_{i=1}^{n} 1 - 1 \\ \text{für } \text{dist}_{nao}(i) > \text{dist}_{ngo}(i) \\ -\frac{1}{2n}\sum_{i=1}^{n} 1 + 1 \\ \text{für } \text{dist}_{nao}(i) < \text{dist}_{ngo}(i) \end{cases} \tag{3.4}$$

$$\begin{aligned} \text{Fitness} &= -\frac{1}{2n}\sum_{i=1}^{n} 2 \cdot (\text{Anzahl der Fehlklassifikationen}) \\ &= -\text{Fehlklassifikationsrate.} \end{aligned} \tag{3.5}$$

D.h. die Fitness ist dann gleich der bisher benutzten Fitness ohne die gewünschte Robustheit. Für größere Werte von q gehen jedoch nicht nur Fehlklassifikationen negativ ein, sondern auch richtig klassifizierte Ansichten, je nachdem wie knapp die Erkennung war. Ist beispielsweise $\text{dist}_{nao}(i) - \text{dist}_{ngo}(i) = 0.2 - 0.1$,

		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	q	b	m	s	b	m	s
	0.00	6.5	8.1	1.2	22.8	26.3	2.4
L=9 neg. KM	0.10	6.9	7.9	0.7	23.2	26.0	1.5
	0.15	7.3	8.3	0.6	23.0	25.3	1.7
	0.20	7.5	8.4	0.7	**22.6**	**25.0**	2.1
	0.25	7.5	8.7	0.7	22.8	25.7	1.8
	0.40	8.3	9.3	0.7	23.8	26.2	1.8

Tabelle 3.6: Ergebnisse der Fehlklassifikationsraten der Optimierung unter Verwendung der direkten Kodierung und der neuen Fitnessfunktion (3.3). Zum einfacheren Vergleich mit der alten Fitnessfunktion sind die entsprechenden Ergebnisse aus Tabelle 3.2 in der ersten Zeile bei $q = 0.00$ aufgetragen. Bei diesem Wert für q ist die Fitnessfunktion identisch mit der bisherigen Funktion. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe.

dann bedeutet das trotz einer richtigen Erkennung (für den Fall von $q = 1$) *noch* einen negativen Beitrag zur Fitness von -0.9, im Gegensatz zu 0.0 bei $q = 0$. Im Falle einer knapp falschen Erkennung von $\text{dist}_{nao}(i) - \text{dist}_{ngo}(i) = 0.1 - 0.2$ bedeutet das *nur* einen negativen Fitnessbeitrag von -1.1, im Gegensatz zu -2.0 bei $q = 0$.

Die Abstandsberechnung erfolgt auf der Ebene der C2-Aktivierungsvektoren $\bar{\mathbf{c}}_2$ und wird nach der folgenden Formel vorgenommen:

$$\text{dist}(\bar{\mathbf{c}}_2(i), \bar{\mathbf{c}}_2(j)) = 1 - \exp(-\frac{||\bar{\mathbf{c}}_2(i) - \bar{\mathbf{c}}_2(j)||^2}{\sigma_{\text{dist}}}). \tag{3.6}$$

Damit ist der Abstand von zwei C2-Aktivierungsvektoren auf den Bereich zwischen 0 und 1 normiert. Bei identischen Vektoren ist der Abstand 0, bei maximal unterschiedlichen Vektoren ist er 1. Der Faktor σ_{dist} skaliert den dynamischen Übergangsbereich. Im Folgenden wird ein Wert von $\sigma_{\text{dist}} = 800$ angenommen, da sich dieser als günstig bei den vorliegenden C2-Aktivierungen erwiesen hat.

Die Ergebnisse der Untersuchung sind in Tabelle 3.6 dargestellt. Die Einstellungen sind identisch zu den bisher verwendeten, mit dem Unterschied, dass nun wieder die COIL20 Datenbank als Datenbank A und die COILselect Datenbank als Datenbank B verwendet wird. Untersucht wird der Fall von $L = 9$ negativen Kombinationsmerkmalen. Außerdem sind zu einer leichteren Veranschaulichung die Mittelwerte der Fehlklassifikationsraten zusätzlich in Abbildung 3.13 dargestellt.

Es ist zu beobachten, dass die Generalisierung 1. Ordnung tendenziell etwas

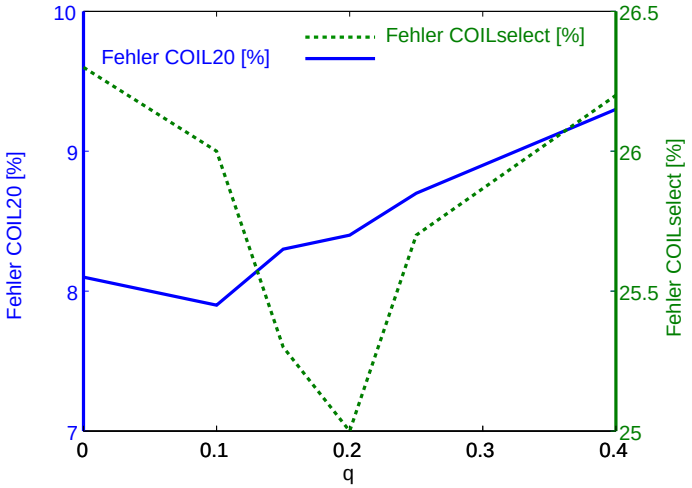

Abbildung 3.13: Mittlere Fehlklassifikationsraten auf COIL20 ($\hat{=}$ negativer Generalisierung 1. Ordnung) und COILselect ($\hat{=}$ negativer Generalisierung 2. Ordnung) bei unterschiedlichen Werten für q.

schlechter ausfällt als bei der evolutionären Optimierung mit reinen Fehlklassifikationen (entspricht $q = 0$) als Fitnessfunktion. Das ist mit der Tatsache zu erklären, dass die Zielsetzung nicht mehr die reine Erkennungsleistung ist. Dem entspricht auch der Verlauf der Erkennungsleistung auf der COIL20. So nimmt bei den Mittelwerten mit fallendem q auch die Fehlklassifikation kontinuierlich ab und wird am Ende ($q = 0.1$) sogar besser als bei der einfachen Fitnessfunktion. Das kann damit begründet werden, dass ein positiver Effekt durch die Glättung der Fitnessfunktion zu erwarten ist. Im Falle von $q = 0$, also der alten Fitnessfunktion, sind die möglichen Fitnesswerte diskretisiert mit der Anzahl der Testansichten. In dem vorliegenden Falle also ist die kleinstmögliche Änderung der Fitness $\frac{1}{24 \cdot 20} = 0.0021$ (20 Objekte mit je 24 Testansichten). Diese Fitnessdifferenz tritt in dem Fall auf, in dem genau eine Testansicht mehr als zuvor korrekt erkannt wird. Diese Diskretisierung wird im Falle für $q > 0$ zugunsten einer beliebig feinen Qualitätsfunktion aufgehoben. Damit ist die Gefahr, dass viele unterschiedliche Individuen, speziell gegen Ende des Optimierungslaufes, die gleiche Fitness aufweisen, vermieden. In einem solchen Fall ist offensichtlicher Weise eine sinnvolle Selektion nicht mehr möglich, was die evolutionäre Optimierung erschwert. Der hier zu erkennende positive Effekt der kontinuierlichen Fitnessfunktion ist jedoch relativ gering. Zu erkennen ist auch, dass die Standardabweichung der Ergebnisse bezüglich der Generalisierung 1. Ordnung kleiner geworden ist. Das könnte auch auf den genannten Glättungseffekt der Fitnessfunktion zurückzuführen sein.

In Bezug auf die Generalisierung 2. Ordnung kann man einen klaren Trend zur Steigerung dieser durch die neue Fitnessfunktion erkennen. Die stärkste Verbesserung von im Durchschnitt 1.3% ist bei einem Wert von $q = 0.2$ zu

beobachten. Dieses Ergebnis ist verglichen mit der bisherigen Fitnessfunktion nicht mit dem üblichen Signifikanzlevel von 95% signifikant, sondern erst bei einem Level von 88%. Eine Erklärung für dieses Ergebnis könnte sein, dass bei einem q von 0.2 der erzielte Effekt noch nicht groß genug ist und bei einem größeren q die Vorteile einer größeren Robustheit der Erkennung von dem Nachteil eines weniger gut auf Erkennung trainierten Systems überkompensiert wird. So wird nämlich u.U. ein visuelles System X, das eine schlechtere Erkennungsleistung als ein System Y hat, mit einer größeren Fitness versehen, weil die einzelnen Erkennungen robuster sind als die bei System Y.

3.8.2 Optimierung mit veränderlicher Objektdatenbank

Eine weitere Möglichkeit, die Robustheit des visuellen Systems während der evolutionären Optimierung zu erhöhen, könnte darin liegen, nicht wie bisher alle Objekte der Datenbank A gleichzeitig zu verwenden, sondern diese dynamisch während der Optimierung auszutauschen. Das bedeutet, dass das visuelle System in einer gewissen Anzahl von Generationen daraufhin optimiert wird, eine Untermenge der Objekte von Datenbank A zu erkennen und in den darauffolgenden Generationen eine andere Untermenge. Damit wird praktisch eine dynamische Fitnessfunktion eingeführt und die Individuen werden zusätzlich dahingehend optimiert, in einer variablen Umgebung zu operieren, d.h. immer wieder andere Objekte gut erkennen zu können.

Bei der Anwendung dieses Verfahrens sind verschiedene Designaspekte zu betrachten:

1. Wie viele Objekte soll eine Untermenge enthalten? Je weniger Objekte in der Untermenge sind, desto größer kann die Veränderung über den Verlauf der Generationen gestaltet werden. Aber in der Zeit, in der diese Untermenge verwendet wird, kann sich das System jetzt u.U. auch noch stärker auf die kleine Anzahl der Objekte anpassen.

2. Wie viele Generationen soll die Untermenge konstant gehalten werden? Je kürzer die Zeit ist, desto geringer ist die Chance des visuellen Systems, sich gezielt an die verwendeten Objekte anzupassen. Gleichzeitig aber wird auch die evolutionäre Suche bei einer zu raschen Änderung der Fitnessfunktion nachhaltig gestört.

3. Um wie viele Objekte soll sich die Untermenge in einem Wechsel verändern? Diese Frage ist eng verknüpft mit der Frage nach der Größe der Untermenge. Wenn die Untermenge sehr groß ist, kann die Änderung von Mal zu Mal auch nur wenige Objekte umfassen. Zu bemerken ist, dass eine Änderung um viele Objekte die „Störung" der Fitnessfunktion vergrößert.

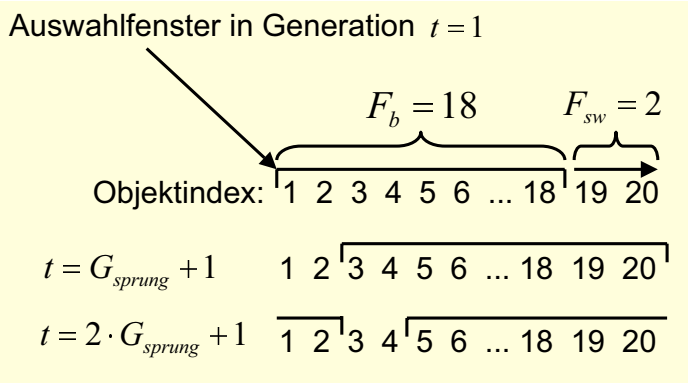

Abbildung 3.14: Schematische Darstellung der Variation der Fitnessfunktion durch die Verwendung nur der Objekte der Datenbank A, die in der jeweiligen Generation in einem Auswahlfenster sind. Die Breite des Auswahlfensters ist mit F_b, die Sprungweite mit F_{sw} und die Verweildauer des Fensters mit G_{sprung} bezeichnet.

4. Soll die Gestalt der einzelnen Objekte einer Untermenge bewusst vom Designer ausgewählt werden, um eine möglichst breite Palette abdecken zu können? Ein Problem bei dem Wechseln der Objekte liegt darin, dass unterschiedliche Objekte über eine unterschiedliche Schwierigkeit in der Erkennung verfügen. Weiter ist vorstellbar, dass Objekte mit einer größeren Ähnlichkeit bei einem Austausch eine kleinere Störung in der Fitnessfunktion verursachen als sehr unterschiedliche Objekte.

Basierend auf den genannten Punkten wird folgendes Schema zur Variation der verwendeten Objekte der Datenbank A eingeführt: Über die 20 Objekte der Datenbank A (COIL20) wird ein „Auswahlfenster" der Breite F_b nach G_{sprung} Generationen um eine Weite von F_{sw} Objekten weiterbewegt. Die Fitnessfunktion verwendet bei der Ermittlung der Klassifikationsleistung stets nur die im Auswahlfenster befindlichen Objekte. Eine schematische Darstellung zu dieser Methode der „Dynamisierung" der bisher statischen Fitnessfunktion findet sich in Abbildung 3.14.

Mit Hilfe der entsprechenden zeitlich veränderlichen Fitnessfunktion werden verschiedene Optimierungsläufe durchgeführt. Die verwendeten Einstellungen entsprechen denen des letzten Abschnittes. Um allein den Effekt der zeitlich veränderlichen Fitnessfunktion zu untersuchen, wird als Fitnessfunktion ansonsten die reine Klassifikationsleistung verwendet (d.h. $q = 0$).

In den Optimierungen von Kapitel 3 war das primäre Ziel die Verbesserung der Generalisierung 1. Ordnung und das sekundäre Ziel die Generalisierung 2. Ordnung. Letztere sollte durch die Optimierung des primären Ziels ebenfalls gesteigert werden. Im Gegensatz dazu ist das Ziel dieses Abschnittes primär die Steigerung der Generalisierung 2. Ordnung. Dies soll durch eine Steigerung der Robustheit der Erkennung innerhalb der Generalisierung 1. Ordnung

erreicht werden. Die Generalisierung 1. Ordnung selbst ist nicht weiter ein Ziel, sondern ein Mittel. In den Optimierungen, in denen die Generalisierung 1. Ordnung das primäre Ziel war, wurde das optimierte visuelle System nicht notwendigerweise der letzten Generation entnommen, sondern der Generation, in der die beste Fitness erreicht wurde. Jetzt ist die Fitness selbst nicht mehr das primäre Ziel, sondern Mittel der Herausbildung eines robusten Erkennungssystems. Dazu wird die Fitnessfunktion dynamisch innerhalb des Optimierungslaufes verändert, wobei auch die Schwierigkeit stark variiert. Wird beispielsweise die Anzahl ähnlicher (und deshalb vom System leicht zu verwechselnder) Objekte in dem Auswahlfenster kleiner, so sinkt die Schwierigkeit der Erkennungsaufgabe. Ein System, das also innerhalb des Optimierungslaufes die minimale Fitness erreicht, ist deshalb weniger das robusteste System, sondern vielmehr das beste System bei einer einfachen Erkennungsaufgabe. Die robustesten Systeme können jedoch am Ende des dynamischen Optimierungslaufes vermutet werden. Um die Auswahl aber auch hier nicht abhängig von dem momentan verwendeten Auswahlfenster zu machen, werden in den letzten G_{sprung} Generationen wieder alle Objekte der Datenbank A verwendet.

Es wäre natürlich auch möglich, die Generalisierung 2. Ordnung in jeder Generation zu messen, diese Information aber nicht in die Fitness mit eingehen zu lassen. Das Problem dieser Vorgehensweise ist jedoch der damit einhergehende erheblich gesteigerte Rechen- und damit Zeitaufwand. Außerdem wäre auch ein direkter Vergleich der erzielten Ergebnisse mit den bisherigen Optimierungsläufen nicht mehr möglich.

Wie bisher üblich, werden 10 unabhängige Optimierungsläufe für unterschiedliche Einstellungen von F_b, F_{sw}, G_{sprung} durchgeführt. Hierbei wird zunächst versucht, möglichst viele Objekte gleichzeitig in dem Auswahlfenster zu haben, um so die Erkennung weiterhin auf eine möglichst breite Basis zu stellen. D.h., F_b wird nahe 20 gehalten. Die Stärke der Veränderung wird im Wesentlichen durch eine behutsame Variation der Verweildauer G_{sprung} und der Sprungweite F_{sw} (eng gekoppelt mit F_b) realisiert. Die Ergebnisse der Untersuchung sind in Tabelle 3.7 dargestellt.

Zur Untersuchung der Auswirkung der Verwendung der dynamischen Fitnessfunktion auf die Generalisierung 1. und 2. Ordnung wird das Szenario mit $L = 9$ negativen Kombinationsmerkmalen verwendet. Außerdem werden zum einfacheren Vergleich wieder die Ergebnisse der statischen Fitnessfunktion (vgl. Tabelle 3.2) in der ersten Zeile bei $G_{sprung} = 400$ aufgeführt (Eine Parametrisierung, die die statische Fitnessfunktion realisiert, ist: $F_b = 20, F_{sw} = 0$ und $G_{sprung} = 400 =$ Generation, bis zu der die Optimierung durchgeführt wird).

Es ist festzustellen, dass die Generalisierungsleistung 1. Ordnung durchweg geringer ist (Fehler auf COIL20 ist größer) als im Falle einer statischen Fitnessfunktion. Das ist ein zu erwartendes Ergebnis, da zum einen die Optimierung immer wieder durch die Veränderung der Fitnessfunktion mittels einer veränderten Objektauswahl gestört wird. Zum anderen werden im Ge-

L=9 neg. KM		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	G_{sprung}	b	m	s	b	m	s
	400	6.5	8.1	1.2	22.8	26.3	2.4
$F_b = 18$	10	8.1	9.3	1.0	24.2	26.3	1.8
$F_{sw} = 2$	20	7.3	9.3	1.2	23.6	25.7	3.0
	30	7.7	9.5	1.0	23.3	25.3	2.3
	40	8.1	9.4	0.7	24.3	26.1	2.0
$F_b = 17$	10	8.3	9.7	0.8	**22.7**	25.4	2.2
$F_{sw} = 3$	20	7.5	9.9	1.2	23.5	25.9	2.4
	30	7.3	9.3	1.1	24.3	25.8	1.6
	40	8.5	9.3	0.6	23.3	**24.8**	1.1
$F_b = 16$	10	8.3	9.7	0.7	24.7	25.8	1.3
$F_{sw} = 4$	20	7.9	9.5	0.9	23.0	25.5	1.7
	30	7.9	9.1	0.7	23.4	25.2	1.7
	40	8.3	9.3	0.7	23.5	25.6	2.3

Tabelle 3.7: Ergebnisse der Optimierung unter Verwendung der zeitlich veränderlichen Fitnessfunktion bestimmt durch das in Abbildung 3.14 dargestellt Objektauswahlschema. Zum einfacheren Vergleich mit der alten Fitnessfunktion ist das korrespondierende Ergebnis aus Tabelle 3.2 mit statischer Fitnessfunktion in der ersten Zeile bei $G_{Sprung} = 400$ aufgetragen. F_b=Breite des Auswahlfensters, F_{sw}=Sprungweite, G_{sprung}=Verweildauer des Fensters, L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe.

gensatz zu den Optimierungen mit der statischen Fitnessfunktion auch nur in den letzten G_{sprung} Generationen *gleichzeitig alle* Objekte der Datenbank A zur Fitnessbestimmung verwendet.

Die Generalisierungsleistung 2. Ordnung hingegen zeigt im Mittel, bis auf ein gleich gutes, immer ein besseres Ergebnis. Damit wird die Vermutung bestätigt, dass das visuelle System durch die Veränderlichkeit der Umgebung zu einem gewissen Grade darauf trainiert werden kann, eine noch allgemeinere Erkennungshierarchie aufzubauen als im bisherigen Falle. Das beste mittlere Ergebnis wird für eine Fensterbreite von $F_b = 17$ und eine Sprungweite von $F_{sw} = 3$ bei einer Verweildauer des Auswahlfensters von $G_{sprung} = 40$ Generationen erreicht. Das hier erreichte mittlere Ergebnis, von 24.8% Fehlklassifikationsrate auf der Datenbank B, ist statistisch signifikant besser als für den Fall einer statischen Fitnessfunktion.

Zur Erreichung dieser erhöhten Robustheit darf die Änderung der Fitnesslandschaft somit nicht zu klein ($F_b = 18, F_{sw} = 2$) und nicht zu groß ($F_b = 16, F_{sw} = 4$) sein. Bei einer zu kleinen Änderung ist offensichtlich der positive Effekt zu gering, und bei einer zu großen Änderung wird der Optimierungsprozess insgesamt zu stark gestört, als dass bei der Generalisierung 2. Ordnung größere Verbesserungen zu erzielen sind. Die Vergrößerung der Verweildauer (G_{sprung}) des Auswahlfensters verringert die Zeitpunkte zu denen die Fitnessfunktion einer Veränderung unterworfen wird und verlängert zugleich die Zeit der Evolution zur ungestörten Optimierung des visuellen Systems. Hier erweist sich – im Rahmen der getesteten Parametrisierung – eine relativ lange Zeitspanne von $G_{sprung} = 40$ als optimal.

Im Folgenden soll exemplarisch der Zeitverlauf der Optimierung, die zu dem System mit der besten Generalisierung 2. Ordnung (Fehler auf COILselect von 22.7%) geführt hat, dargestellt werden. Die Einstellungen für diesen Lauf waren: $F_b = 17, F_{sw} = 3, G_{sprung} = 10$. Das bedeutet, dass alle 10 Generationen das 17 Objekte umfassende Auswahlfenster um 3 Objektindices weiter verschoben wird[7]. Dadurch ergibt sich – wie bereits erwähnt – eine periodisch veränderliche Schwierigkeit der Erkennungsaufgabe. Zur Bewertung der jeweiligen Schwierigkeit wurde die Erkennungsleistung des besten visuellen Erkennungssystems der entsprechenden statischen Optimierung (6.5% Fehler auf COIL20), verwendet. Um einen einfacheren Vergleich zu gewährleisten, wurde der Fehlklassifikationsleistung des Referenzsystems ein konstanter Wert von 0.04 hinzuaddiert. Der zeitliche Verlauf der Fehlklassifikationsleistung des besten Individuums ist in Abbildung 3.15 dargestellt. Außerdem gezeigt ist die beste erreichte Fehlerrate und der Verlauf der Schwierigkeit der Klassifikationsaufgabe.

Man kann erkennen, dass die Periodizität der Fehlklassifikationsrate des

[7]Nachdem der Startindex des Fensters die Nummer 20 überschritten hat, beginnt das Fenster wieder bei dem Objekt Nummer eins.

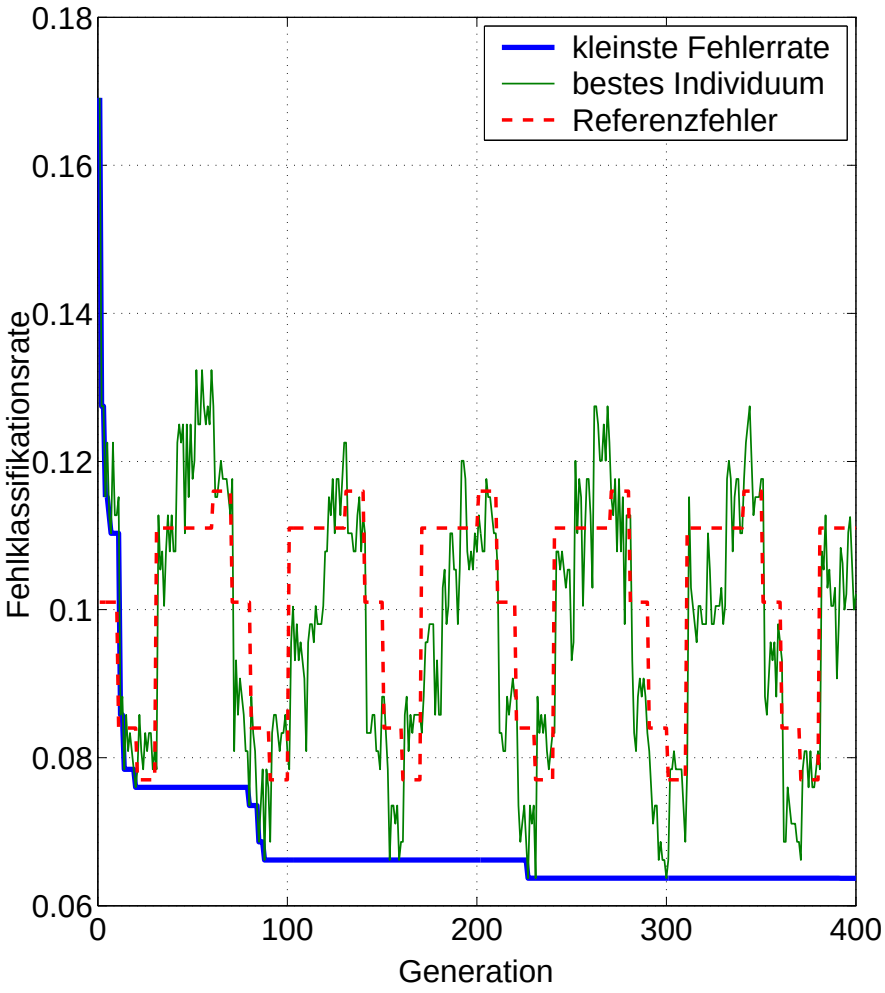

Abbildung 3.15: Verlauf der Fehlklassifikationsrate des besten Individuums während der Optimierung der zeitlich veränderlichen Fitnessfunktion („–"). Beste je erreichte Fehlklassifikationsrate („-"). Verlauf der Schwierigkeit der Klassifikationsaufgabe berechnet durch Referenzsystem („- -") (zum besseren Vergleich um 0.04 angehoben).

besten Individuums in groben Zügen mit der des Referenzsystems („Referenzfehler") übereinstimmt. Eine genauere Übereinstimmung ist aus zwei Gründen nicht zu erwarten: Zum einen schwankt die Schwierigkeit, die ein individuelles visuelles System bei einer bestimmten Auswahl an Objekten hat. Zum anderen liegen im Falle der dynamischen Variation der Fitnessfunktion nur begrenzte Anpassungsmöglichkeiten vor, was leichte Veränderungen in der Schwierigkeit bestimmter Aufgaben hervorruft. So kann zwar eine bestimmte Objektmenge durchaus einfacher zu erkennen sein, das dafür notwendige System benötigt jedoch dafür eine spezielle Kombinationsmerkmalseinstellung, die allerdings für alle übrigen Objektmengen nicht von Nutzen ist. Eine solche Überanpassung wird durch die veränderliche Umgebung vermindert und damit ändern sich auch die Schwierigkeiten der einzelnen Erkennungsaufgaben.

Weiter ist in Abbildung 3.15 zu erkennen, dass nach einem anfänglich schnellen Absinken der Fehlklassifikationsrate die „Schwingung" nur noch sehr langsam weiter absinkt. Dieser sehr langsame weitere Abfall ist für den Tiefpunkt der Schwingung zu beobachten. Das bedeutet, dass ab ca. Generation 100 – nach einer vollen Periode – nur noch kleine Verbesserungen der Robustheit des Systems erreicht wurden.

Kapitel 4

Untersuchung des visuellen Systems und des Entwurfsverfahrens

Ein fundamentales Problem bei der evolutionären Optimierung von Systemen liegt darin, festzustellen, ob das gefundene Optimum das globale oder lediglich ein lokales ist. Insbesondere in den Fällen, in denen das optimierte System die gestellte Aufgabe nicht zu 100% löst, stellt sich die Frage, ob es überhaupt eine solche Einstellung für das vorliegende System gibt oder ob das System mit den vorliegenden Randbedingungen keine 100%ige Lösung des Problems erlaubt. Es sind also zumindest zwei Fragen, die hier ineinander greifen:

1. Wie leistungsfähig ist die evolutionäre Optimierung des Systems?

2. Wie leistungsfähig ist das System in der bestmöglichen Konfiguration?

Mit der zweiten Frage in engem Zusammenhang steht die Frage: Wie viele lokale Optima gibt es für das System und wie hängen diese zusammen? Diese Fragen sollen im folgenden Kapitel nun von einer anderen Seite als bisher in dieser Arbeit untersucht werden. Um die Funktionsweise und den Konfigurationsraum des visuellen Systems und das Zusammenspiel mit der evolutionären Optimierung besser verstehen zu können, soll ein im Folgenden generativ genanntes Modell vorgestellt werden.

Dieses generative Modell soll dazu dienen, eine Objektdatenbank – oder genauer eine Musterdatenbank – zu generieren, die dem visuellen System als Erkennungsaufgabe präsentiert wird. Dabei sollen die Muster verschiedene Eigenschaften erfüllen. Zum einen soll das darauf aufbauende Klassifikationsproblem so beschaffen sein, dass zumindest eine optimale Lösung explizit bekannt sei. Zum anderen soll das Problem noch schwer genug sein, dass es eine Herausforderung für System und Optimierung darstellt. Aus diesem Grund soll die Klassifikationsaufgabe nicht linear separabel sein und damit mit Hilfe

eines einfachen SLPs nicht lösbar sein. Weiter wird die Annahme getroffen, dass die einzelnen Muster in einer hierarchischen Weise aufgebaut sein sollen. Diese Annahme entspricht der Auffassung, dass auch reale Objekte einen hierarchischen Aufbau aufweisen und daher auch der Ansatz eines hierarchischen Erkenners von Vorteil bei der Objekterkennung ist. Dazu werden die Ergebnisse des visuellen Systems, die mit der annähernd gleichen Optimierung wie im Kapitel zuvor optimiert werden, mit denen eines MLPs verglichen.

Im folgenden Abschnitt wird das generative Modell zunächst prinzipiell erklärt und im Anschluss daran an einem Beispiel verdeutlicht. Die nachfolgenden Analysen arbeiten auf dem beispielhaft erzeugten hierarchischen Mustererkennungsproblem.

4.1 Generatives Modell

Das generative Modell kann in gewisser Weise als die Inversion einer starken Vereinfachung des vorgestellten hierarchischen visuellen Systems angesehen werden[1]. Bei dem visuellen System werden im Inputbild, das das visuelle Feld repräsentiert, einfache optische Merkmale detektiert. In einem nächsten Schritt wird das räumliche Pooling durchgeführt. Das bedeutet, dass die Information über die exakte Lage eines Merkmals verloren geht. Lediglich die Information über eine grobe Lage im visuellen Feld bleibt erhalten. In dem folgenden Schritt werden die Kombinationsmerkmale detektiert. Diese Merkmale höherer Ordnung entstehen durch die räumliche Kombination einfacher Merkmale. Im Anschluss daran wird auch die räumliche Information über die Kombinationsmerkmale vergröbert. Die anschließende Objekterkennung basiert dann auf der so transformierten Information. Durch die Verwendung des räumlichen Poolings wird es so möglich, unterschiedliche Ansichten eines Objektes zu erkennen, ohne dass jede mögliche Merkmalsanordnung einer Ansicht explizit festzuhalten ist. Vielmehr wird nur eine ungefähre Lage von einfachen und komplexeren optischen Merkmalen im Raum und zueinander repräsentiert[2]. Der prinzipielle Aufbau der hierarchischen Objekt- bzw. Mustererkennung ist in Abbildung 4.1 dargestellt. Die Umkehrung dieses Aufbaus entspricht dem prinzipiellen Aufbau des generativen Modells.

Das generative Modell, das jetzt eine Musterdatenbank erzeugen soll, geht genau den umgekehrten Weg. Statt der Konvergenz der Informationen zu je einer Objektrepräsentation, wird je **eine** Musterrepräsentation in **viele** unterschiedliche Muster einer Musterklasse verteilt. Dies wird durch eine inverse Form des räumlichen Poolings erreicht. Dabei wird je ein Muster für eine mögliche räumliche Position eines Kombinationsmerkmals oder eines einfachen Merkmals generiert. Damit erhält man aus der Repräsentation einer

[1] Das tatsächliche hierarchische visuelle System kann aufgrund der verwendeten Nichtlinearitäten nicht invertiert werden.

[2] Das Problem der kombinatorischen Explosion wird dadurch abgemildert.

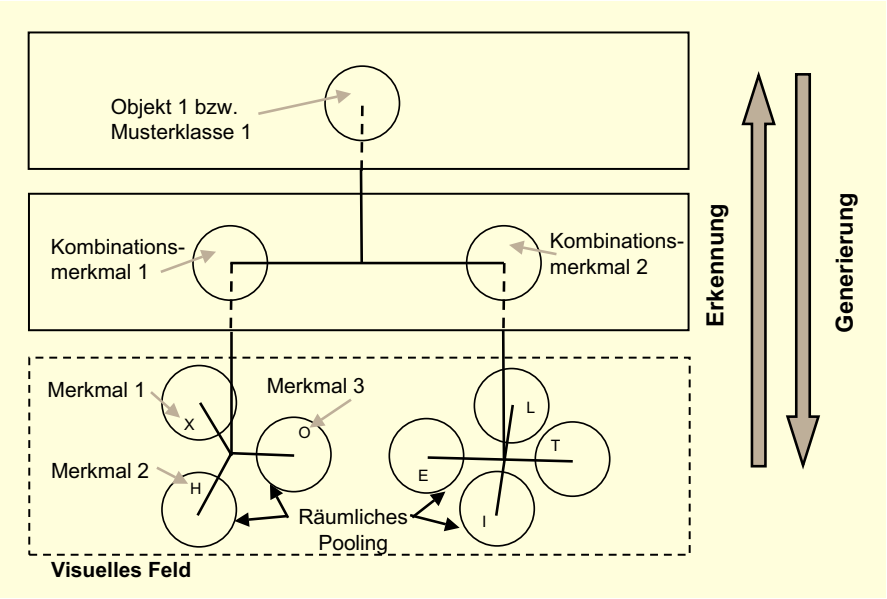

Abbildung 4.1: Prinzipieller Aufbau der Objekterkennung mit Hilfe des visuellen Systems. Durch die Umkehrung dieses Aufbaus wird das generative Modell realisiert. Bei diesem werden aus **einer** Objekt- oder Musterrepräsentation **viele** unterschiedliche Muster einer Musterklasse generiert.

Musterklasse eine Vielzahl von Mustern, die sich aus allen möglichen räumlichen Kombinationen der Merkmale und Kombinationsmerkmale innerhalb der Poolingbereiche ergeben.

Der Zustandsraum aller möglichen Muster sei mit O bezeichnet. Der Zustandsraum teilt sich nicht-überlappend in O_i Unterräume auf, wobei in O_i alle unterschiedlichen Beispiele der Musterklasse i enthalten sind. Die unterschiedlichen Beispiele eines Musters sind analog zu den unterschiedlichen Ansichten eines Objektes zu sehen. Nicht betrachtet wird der Fall, dass sich mehrere Muster in einem Inputbild befinden. Die Nichtexistenz eines Musters im Input kann als eine getrennte Musterklasse betrachtet werden. Eine Teilmenge von O ist T, welche die Menge der Inputs umfasst, die dem visuellen System während einer Trainingsphase, zusammen mit der entsprechenden Labelinformation, gezeigt werden. T ist ebenso wie O in nicht-überlappende Unterräume T_i aufgeteilt, wobei der Teilraum T_i die Trainingsansichten des Objektes i enthält. N_{O_i} und N_{T_i} bezeichnen die Anzahl der Elemente der Unterräume O_i und T_i. Im Folgenden wird aus Gründen der Einfachheit angenommen, dass gilt: $N_{O_i} = N_{O_j}$, $N_{T_i} = N_{T_j}$ für alle i, j. Weiter gelte $N_{O_i} > N_{T_i}$.

Im folgenden Abschnitt wird nun aufbauend auf den erläuterten Prinzipien des generativen Modells exemplarisch eine konkrete Musterdatenbank generiert. Diese wird nachfolgend dazu verwendet, die Leistungsfähigkeit des visuellen Systems und des evolutionären Entwurfsprozesses zu untersuchen.

4.2 Generierte Musterdatenbank

In diesem Abschnitt wird zunächst ein sehr einfaches generatives Modell vorgestellt. Es besteht aus zwei Musterklassen, von denen jede $N_{O_1} = N_{O_2} = 243$ Muster enthält. Jede Klasse wird durch genau ein Kombinationsmerkmal – K_1 bzw. K_2 – eindeutig definiert. Beide Merkmale bauen sich jeweils aus den zwei einfachen Merkmalen $\mathbf{M}_1$ und $\mathbf{M}_2$ auf. Die beiden einfachen Merkmale $\mathbf{M}_1$ und $\mathbf{M}_2$ haben eine räumliche Ausdehnung von 3×3 Pixel und sind durch folgende Matrizen definiert:

$$\mathbf{M}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \tag{4.1}$$

$$\mathbf{M}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{4.2}$$

Eingebettet sind diese beiden Merkmale in ein mit Nullen initialisiertes visuelles Feld der Größe 20×5 Pixel. Kombinationsmerkmal K_1 ist durch die räumliche Kombination des Merkmals $\mathbf{M}_1$ **über** dem Merkmal $\mathbf{M}_2$ definiert. Umgekehrt ist K_2 durch die räumliche Kombination des Merkmals $\mathbf{M}_1$ **unter** dem Merkmal $\mathbf{M}_2$ definiert. Durch das inverse Pooling der Merkmale $\mathbf{M}_1$ und $\mathbf{M}_2$ um ein Pixel in alle Raumrichtungen können die 3×3 Pixel großen Merkmale neun unterschiedliche Positionen innerhalb eines 5×5 Pixel großen Bereiches innerhalb des Inputbildes einnehmen. Das inverse Pooling der Kombinationsmerkmale umfasst drei unterschiedliche Positionen. Durch dieses generative Modell ergeben sich insgesamt 486 unterschiedliche Muster ($2 \cdot 3 \cdot 9 \cdot 9 = 486$) $N_{O_1} = N_{O_2} = 243$ für jede Musterklasse. Das entsprechende generative Modell ist rein schematisch in Abbildung 4.2 dargestellt.

Zu beachten ist, dass der nicht von Mustern belegte Bereich nach der Verteilung der Merkmale mit dem Wert 0 belegt wird. Eine zufällige Auswahl von unterschiedlichen Mustern beider Klassen, die mit dem vorgestellten generativen Modell erzeugt wurden, ist in Abbildung 4.3 dargestellt.

Die generierte Musterdatenbank soll jetzt an die Stelle der COIL20 treten und das zu lösende Klassifikationsproblem darstellen. Um zu verhindern, dass das Problem der Trennung der beiden Klassen zu einfach ist, sind zwei Umstände notwendig. Zum einen ist zu gewährleisten, dass die beiden Klassen nicht durch die räumliche Lage eines einfachen Merkmals zu unterscheiden sind. Aus diesem Grunde gibt es einen genügend großen Überschneidungsbereich, in dem beide Merkmale $\mathbf{M}_1$ und $\mathbf{M}_2$ auftreten und zwar je innerhalb von Klasse 1 und Klasse 2. Zum anderen ist zu gewährleisten, dass kein einzelnes großes Merkmal existiert, das beide Klassen voneinander trennen kann. Ein solches Merkmal könnte man sich z.B. aus der kompletten Variation eines Objektes

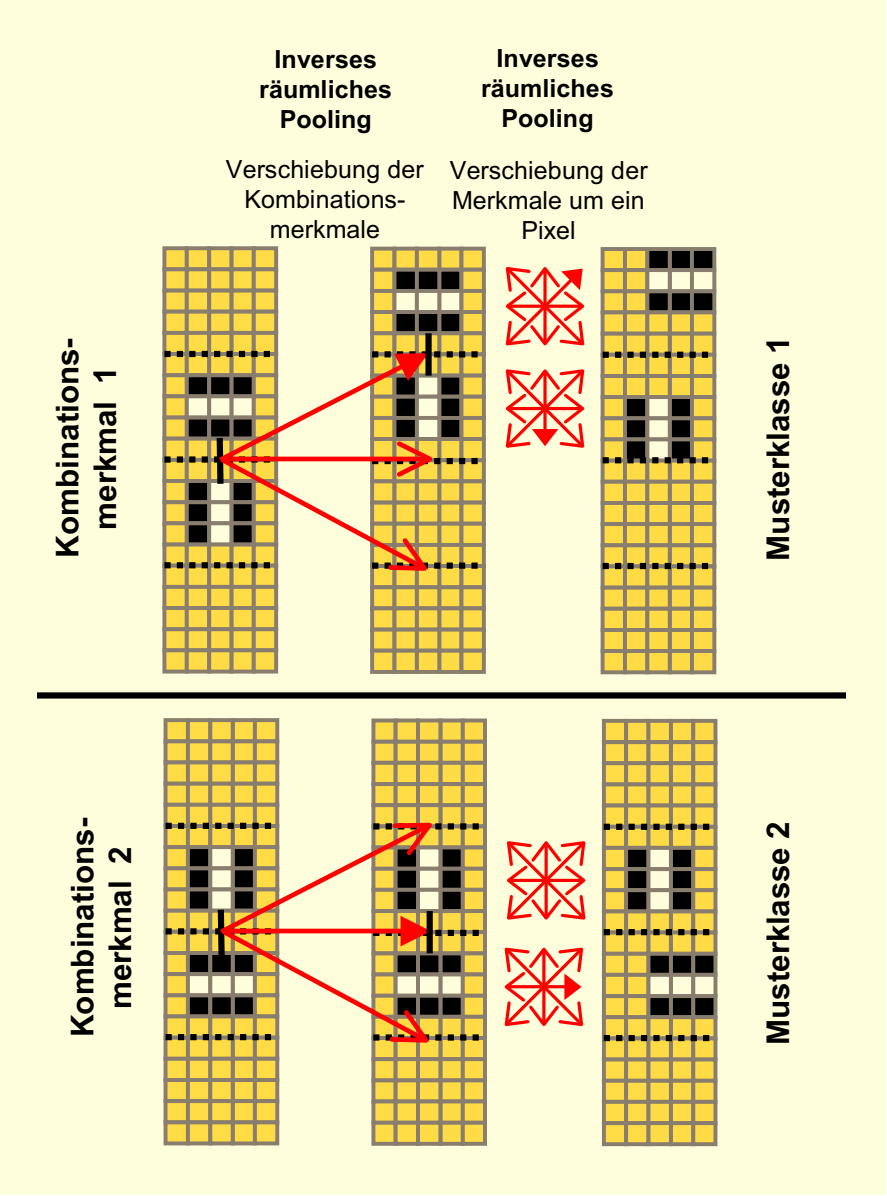

Abbildung 4.2: Schematische Darstellung eines exemplarischen generativen Modells. Aufbauend auf je einem unterschiedlichen Kombinationsmerkmal (bestehend aus je zwei einfachen Merkmalen) werden durch hierarchisches inverses Pooling je 243 unterschiedliche Muster einer Musterklasse erzeugt. Die Muster mit einer „T"-artigen Struktur gehören zu der Klasse 1. Die Muster der Klasse 2 haben die Struktur eines auf dem Kopf stehenden „T". Bei den Merkmalen $\mathbf{M}_1$ und $\mathbf{M}_2$ ist der Wert 1 mit der Farbe Weiß und der Wert 0 mit der Farbe Schwarz symbolisiert.

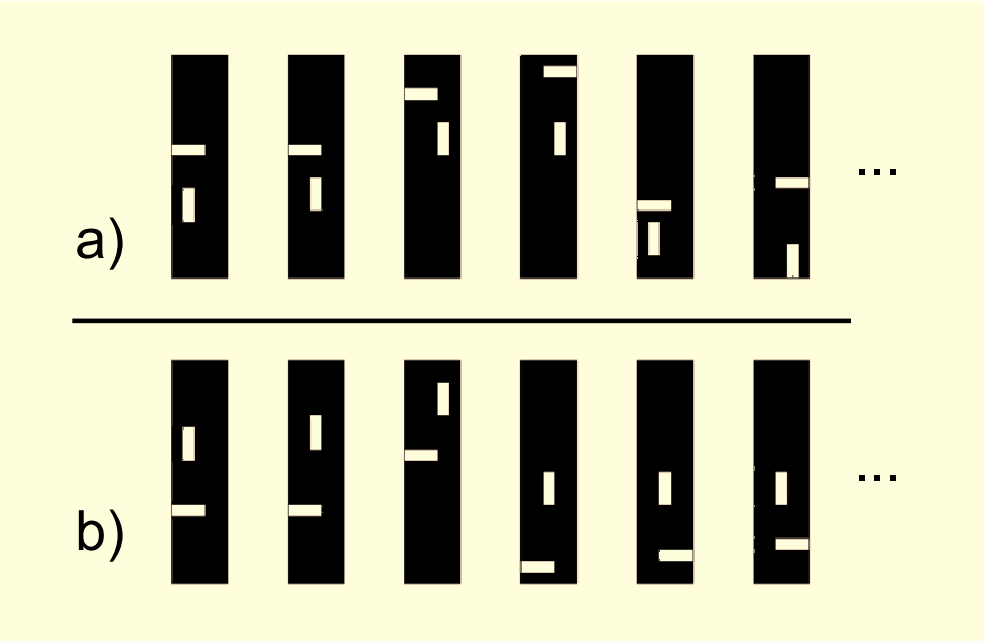

Abbildung 4.3: Zufällige Auswahl von je 6 (der 243) Muster von a) Musterklasse 1 und b) Musterklasse 2. Der Pixeleintrag 1 ist mit der Farbe Weiß und der Eintrag 0 mit der Farbe Schwarz symbolisiert.

im Ansichtsraum aufgebaut vorstellen. Da sich jedoch die einzelnen Variationen von Klasse zu Klasse und von Merkmal zu Merkmal stark überlappen, ist ein solches Merkmal nicht existent. Anders ausgedrückt kann man auch sagen: Die beiden Klassen sind nicht linear separabel. D.h., ein Single-Layer-Perceptron (SLP) kann die beiden Klassen nicht vollständig korrekt trennen. Um dieses jedoch zu beweisen und nicht aus der nicht erfolgten Konvergenz einer SLP-Lernregel zu schließen, dass die Klassen nicht linear trennbar sind, wurde ein entsprechendes Beweisverfahren angewandt. Dieses basiert auf der Anwendung des Schemas der Linearen Programmierung und ist in [6] Seite 258 ff. beschrieben. Die Anwendung des Verfahrens auf die generierte Musterdatenbank erbrachte den Beweis ihrer Nichtseparabilität.

Ein Multi-Layer-Perceptron (MLP), das hingegen mit einer genügend großen Anzahl von versteckten Knoten ausgerüstet ist, ist in der Lage, die beiden Klassen perfekt zu trennen, da es jede beliebige Funktion approximieren kann. Die jedoch interessantere Frage ist: Wie gut wird ein solches Netz generalisieren? Eine entsprechende Untersuchung folgt im nächsten Abschnitt.

4.2.1 Klassifikation mit Multi-Layer-Perceptron

Um die Generalisierungsfähigkeit von MLPs bei der generierten Erkennungsaufgabe zu messen, wurden zufällig 52 Ansichten je Musterklasse als Trainingsmuster ausgewählt. Damit ist $N_{O_1} = N_{O_2} = 243$ und $N_{T_1} = N_{T_2} = 52$. Die

verwendete Architektur der angewandten MLPs sieht folgendermaßen aus: Die Eingangsschicht besteht aus 100 Neuronen, entsprechend der 100 Pixel großen Inputbilder ($20 \times 5 = 100$). Für die folgende Schicht von versteckten Neuronen wurden verschiedene Szenarien getestet: 30, 20, 10 und 5 versteckte Neuronen mit einer sigmoidalen Transferfunktion. Jede Netzstruktur wurde 10-mal mit unterschiedlichen Zufallsinitialisierungen trainiert. Die Ausgangsschicht enthält nur ein Neuron mit einer linearen Transferfunktion. Die Verschaltung der Neuronen ist vorwärtsgerichtet ohne „Shortcuts". Die Netze wurden darauf trainiert, bei der Präsentation eines Musters der Klasse 1 eine 1.0 und bei der Präsentation eines Musters der Klasse 2 eine -1.0 auszugeben. Das Training wurde in allen Fällen mit Hilfe des Levenberg-Marquardt-Verfahrens durchgeführt [10], und zwar so lange, bis die Netze eine 0%-ige Fehlklassifikation auf den Trainingsdaten erreichten. Dies war im Allgemeinen innerhalb von 300 Lernschritten erreicht[3].

Die Fehlklassifikationsrate der trainierten Netze wird folgendermaßen berechnet: Wird nach der Präsentation eines Inputbildes am Ausgangsneuron ein positiver Wert angenommen, so wird das Bild vom Netz als ein Muster der Klasse 1 klassifiziert, für einen Ausgangswert kleiner als Null als ein Muster der Klasse 2. Anschließend wird der prozentuale Fehler, die Fehlklassifikationsrate, berechnet. Bei diesem Fehler geht damit nicht ein, wieweit der Ausgangswert von den zu lernenden Werten 1.0 und -1.0 entfernt ist, sondern nur, ob die Entscheidung basierend auf dem Ausgangswert falsch oder korrekt ist.

Die Bewertung der trainierten neuronalen Netze erfolgt anhand der Fehlklassifikationsrate auf den verbleibenden 382 Ansichten ($486 - 104 = 382$), den Testbildern. Über 10 Trainingsläufe gemittelt erreichte das Netz mit 10 versteckten Neuronen die beste Erkennungsleistung auf den Testbildern. Die mittlere Fehlklassifikationsrate betrug 11.8% bei einer minimalen Rate von 6.0% und einer Standardabweichung von 4.3%. Im Vergleich dazu wird im folgenden Abschnitt die Erkennungsleistung des visuellen Systems auf den generierten Daten untersucht.

4.2.2 Optimierung des visuellen Systems

Die Erkennung der generierten Daten erfolgt im Wesentlichen genau wie bei der Erkennung der COIL20 oder COILselect Bilder. Auch hier wird das visuelle System nur dadurch trainiert, dass lediglich je drei der Trainingsansichten als Repräsentanten für ein Objekt bzw. eine Musterklasse als C2-Aktivierungen abgespeichert werden. Die anschließende Erkennung oder Klassifikation der präsentierten Bilder erfolgt, wie bereits zuvor beschrieben, durch die Auswertung der Nächsten-Nachbar-Abstände.

[3]Es wurden außerdem verschiedene Läufe mit einem Early-Stopping-Verfahren trainiert, um ein etwaiges Overfitting der Netze zu verhindern. Diese konnten jedoch die Generalisierungsleistung der MLPs nicht weiter verbessern.

Das visuelle System wird, wie in Kapitel 3 zuvor beschrieben, in einer direkt kodierten evolutionären Suche optimiert. Es werden lediglich positive Einträge für die Kombinationsmerkmale verwendet. Optimiert werden wiederum die Parameter der Nichtlinearitäten und die Kombinationsmerkmale. Im Unterschied zu den Einstellungen des letzten Kapitels wird die Anzahl der Kombinationsmerkmale auf zwei festgelegt, die mindestens notwendige Anzahl zur Lösung des Problems. Das Optimierungsproblem wird damit einfacher, da der Suchraum erheblich eingeschränkt wird. Gleichzeitig wird jedoch durch eine Änderung an einer anderen Stelle die Suche stark erschwert. Und zwar werden die bisher als gegeben angenommenen Gabormerkmale der ersten Schicht zusätzlich variabel kodiert (Die Werte dieser Merkmale werden genau wie die Kombinationsmerkmale auf das Intervall [0,1] beschränkt.). Hierzu werden zwei 3×3 Merkmale für die erste Schicht angenommen, die genau wie die Kombinationsmerkmale bisher behandelt werden. Damit ergibt sich für den evolutionären Suchraum eine Dimension von 60. Diese setzt sich aus den sechs Nichtlinearitäten, den beiden Merkmalen der ersten Schicht ($2 \times 3 \times 3 = 18$ freie Parameter) und den beiden Merkmalen der zweiten Schicht ($2 \times 2 \times 3 \times 3 = 36$ freie Parameter) zusammen. Die Dimension des Suchraumes ist damit zwar kleiner als zuvor, jedoch sind nun die zu bestimmenden Merkmale in zwei Schichten hintereinander angeordnet. Damit wird die Suche weiter erschwert, da eine Änderung der einfachen Merkmale der ersten Schicht auch eine direkte Auswirkung auf die Leistungsfähigkeit der Kombinationsmerkmale der zweiten Schicht hat. Es kommt also zusätzlich zu der Verkopplung der Parameter der Nichtlinearitäten eine weitere zwischen den einfachen und den höheren Merkmalen hinzu.

Die Musterdatenbank, die für die Ermittlung der Generalisierungsleistung 1. Ordnung benutzt wird, besteht aus 49 der 52 Trainingsansichten (je Musterklasse). Diese 49 Muster sind nämlich innerhalb des evolutionären Lernens ebenfalls als Trainingsansichten anzusehen. Zusammen mit den drei Repräsentanten, den Trainingsansichten des visuellen Systems, werden somit insgesamt auch je 52 Trainingsansichten verwendet. Um den Vergleich mit den MLPs stringent zu machen, werden in beiden Fällen die identischen Ansichten eingesetzt. Als Testdaten werden wie im Falle der MLPs die verbleibenden 382 Ansichten verwendet.

Zur evolutionären Optimierung werden die identischen Einstellungen von Kapitel 3 verwendet (eine (7,19)-Strategie über 400 Generationen). Folgende Ergebnisse bezüglich der Fehlklassifikationsrate wurden erreicht: In sechs der 10 durchgeführten Läufe wurde eine durchschnittliche Fehlklassifikationsrate von 40% sowohl auf den Trainings- als auch auf den Testdaten erzielt. Dieser sehr große Fehler zeigt, dass in diesen Fällen die Optimierung des visuellen Systems fehlschlug. Jedoch wurde in den verbleibenden vier der 10 Läufe eine Fehlklassifikationsrate von 0% sowohl auf den Trainings- als auch auf den Testdaten erreicht. Es zeigt sich damit also ein komplett anderes Verhalten als

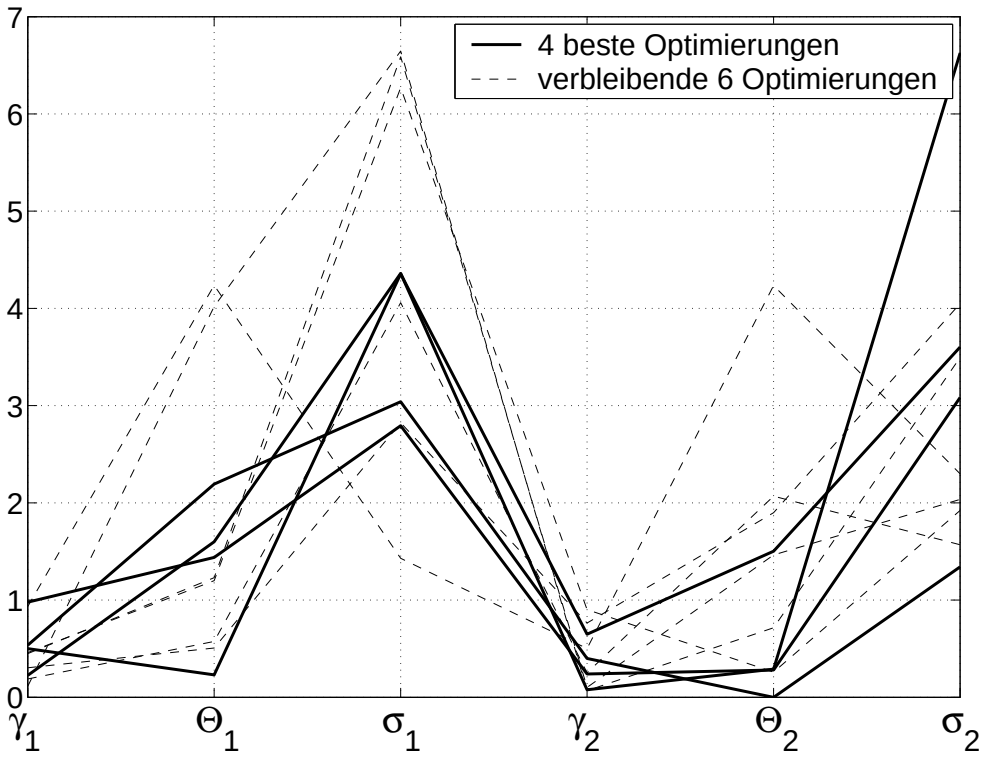

Abbildung 4.4: Werte der Nichtlinearitäten des jeweils besten Individuums eines der 10 Optimierungsläufe auf der generierten Musterdatenbank.

bei den MLPs.

Aufbau der optimierten visuellen Systeme

Anders als bei den Optimierungen auf der COIL Datenbank ist bei der Optimierung auf der generierten Musterdatenbank der Unterschied in der Generalisierungsleistung der 10 visuellen Systeme (je das beste aus einem Optimierungslauf) groß. Die besten vier der 10 Systeme haben eine perfekte Generalisierungsleistung, während die übrigen sechs mit ihren durchschnittlich 40% Fehlklassifikationsleistung eine sehr schlechte Performanz aufweisen. Die Grenze zwischen diesen beiden Gruppen ist stark ausgeprägt. So ist die Fehlklassifikationsleistung des besten der sechs schlechteren Netze immer noch 23%. Es stellt sich die Frage, ob man anhand der gefundenen Nichtlinearitäten erkennen kann, was geeignete und was ungeeignete Parametersätze sind. In Abbildung 4.4 sind die Nichtlinearitäten des besten visuellen Systems je eines Optimierungslaufes dargestellt. Ein gemeinsamer Satz (zugehörig zu einem visuellen System) ist mit Linien verbunden. Die Nichtlinearitäten der vier besten Systeme sind mit durchgezogenen Linien dargestellt, während die verbleibenden sechs mit gestrichelten Linien dargestellt sind. Bei der Betrachtung der Nichtlinearitäten zeigt sich, dass in jedem Optimierungslauf unterschiedli-

che Werte gefunden wurden. Weiter sind weder bei den vier besten Systemen noch bei den sechs schlechteren Systemen Ähnlichkeiten der Parameter der Nichtlinearitäten zu sehen. Das entspricht dem Ergebnis der Optimierung der visuellen Systeme auf die Erkennung der COIL20 Datenbank. In Abbildung 4.4 sind jedoch keine zwei Grundstrategien wie im früheren Fall zu erkennen.

Diese starke Bandbreite der vier das Problem optimal lösenden Nichtlinearitäten ist dadurch zu erklären, dass in dieser Optimierung zusätzlich auch die Merkmale der ersten Schicht optimiert werden. So hängt beispielsweise der erste Schwellwert θ_1 direkt davon ab, wie hoch die durchschnittlichen Werte innerhalb der ersten Merkmale sind. Weiter ist auch der Winner-Take-Most-Parameter γ_1 nun davon abhängig, welche beiden Merkmale sich in der ersten Schicht ausgebildet haben.

Eine Untersuchung der unterschiedlichen Kombinationsmerkmalsbänke mit Hilfe des Abstandsmaßes (siehe Gleichung (3.1) auf Seite 65) wie in Abschnitt 3.7 ist in diesem Fall nicht sinnvoll, da diese insofern schon voneinander verschieden sein müssen, weil sie auf unterschiedlichen Merkmalen der ersten Schicht aufbauen. Eine Untersuchung der Merkmale der ersten Schicht mit Hilfe des Abstandsmaßes ist ebenfalls wenig sinnvoll, da die Merkmale, wenn sie zur Klassifikation der generierten Muster verwendet werden, einen weiteren Verschiebungsfreiheitsgrad haben. Das ist damit zu begründen, dass die Bereiche der künstlichen Muster, die ungleich Null sind, nur eine Breite von einem Pixel aufweisen. Damit sind die 3×3-Pixel breiten Merkmale der ersten Schicht in der Lage, die Linienmerkmale der Bilder mit Linienaktivierungen an unterschiedlichen Stellen im Filter zu detektieren.

Betrachtet man den Aufbau der generierten Muster, so ist zu erwarten, dass gute Merkmalsbänke der ersten Schicht über je ein Merkmal zur Detektion sowohl einer vertikalen als auch einer horizontalen Linie verfügen. Deswegen kann man vermuten, dass die perfekt funktionierenden Erkenner als Merkmale in der ersten Schicht je einen Erkenner für eine horizontale und einen für eine vertikale Linie aufweisen werden. Dieses sind die fundamentalen Bestandteile der beiden vorkommenden Objekte. D.h., man vermutet die Ausbildung von einer horizontalen und einer vertikalen Linie innerhalb der beiden Merkmale. Diese könnten an unterschiedlichen Positionen im Filter auftreten. Betrachtet man jedoch die Merkmalsbänke der vier optimalen Systeme (Abbildung 4.5), so wird diese Vermutung nur zu einem kleinen Teil bestätigt. So ist lediglich im ersten und dritten visuellen System die ansatzweise gleichzeitige Ausprägung von zwei unterschiedlich orientierten Linienmerkmalen zu erkennen (gekennzeichnet mit einer gestrichelten Linie). In dem zweiten und vierten System hingegen sind keine solche Linienmerkmale mehr auszumachen (bis auf ein einzelnes im zweiten Merkmal des zweiten Systems). Somit zeigt sich, dass trotz der Verwendung dieses stark vereinfachten visuellen Systems die Einstellmöglichkeiten, die zu einer erfolgreichen Erkennung führen, immer noch sehr groß sind.

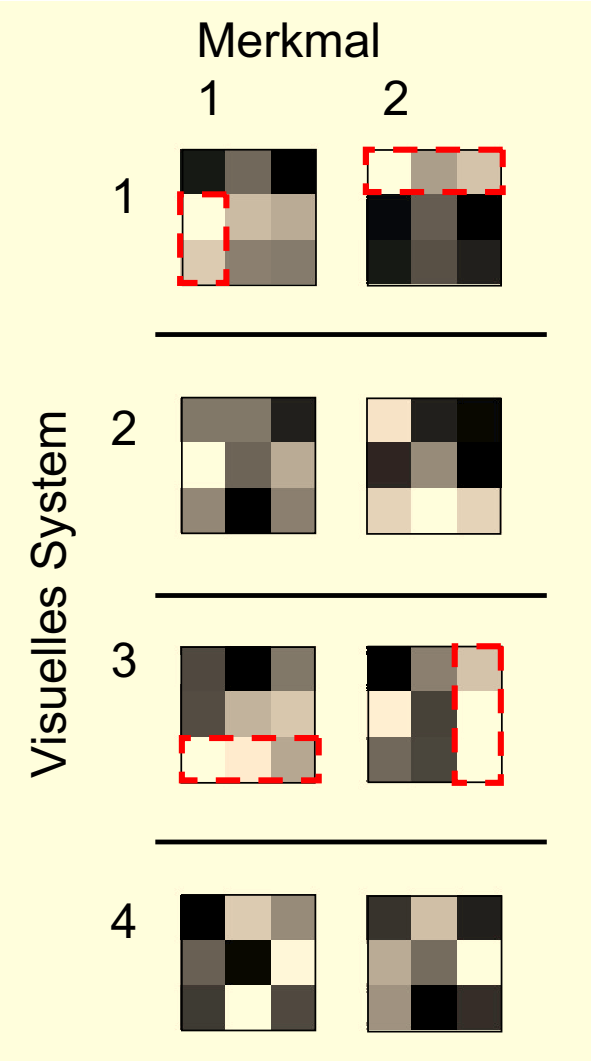

Abbildung 4.5: Optimierte Merkmalsbänke der vier besten visuellen Systeme (diese haben je eine Fehlklassifikationsrate von 0% auf den Testdaten). Lediglich im ersten und dritten visuellen System kann man ansatzweise die gleichzeitige Ausprägung von zwei unterschiedlich orientierten Liniendetektoren (gekennzeichnet mit einer gestrichelten Linie) erkennen. Die Werte der beiden optimierten Merkmale jedes visuellen Systems sind auf das Intervall [0,1] beschränkt. Der Wert 0 ist mit einem schwarzen und der Wert 1 mit einem weißen Pixel dargestellt. Werte zwischen 0 und 1 sind mit unterschiedlich hellen Grautönen dargestellt.

Diskussion

Es kann Folgendes gesagt werden: Das relativ einfach erscheinende Klassifikationsproblem, das mit Hilfe des generativen Modells erzeugt wurde, bereitet schon hinreichende Schwierigkeiten, sowohl beim Training als auch bei der Generalisierung. So zeigen Multi-Layer-Perceptron-Netze im Mittel einen Generalisierungsfehler von 12%. Der evolutionäre Entwurf des visuellen Systems erreicht sogar in 60% der Fälle nur eine sehr schlechte Erkennungsrate auf den Trainingsdaten. In den verbleibenden 40% jedoch erreichen die optimierten Systeme sowohl einen Trainingsfehler als auch einen Testfehler von 0%. Es kommt also zu keinem Auseinanderfallen von Trainings- und Testfehler. Begründet werden kann diese erhöhte Fähigkeit zur Generalisierung durch den biologisch inspirierten hierarchischen Aufbau des Erkennungssystems. Der Vorteil dieses Aufbaus kommt insbesondere bei den durch das generative Modell hierarchisch erzeugten Mustern zum Tragen. Das evolutionäre Lernen kann nicht Ursache der besseren Generalisierungsfähigkeit des visuellen Systems verglichen mit dem MLP sein, da zum einen das evolutionäre Training sowie das Training des MLPs auf einem identischen Satz von Trainingsdaten beruht. Zum anderen zielte das evolutionäre Lernen – genau wie das Backpropagation-Lernen des MLPs – ausschließlich auf die Verbesserung der Erkennungsrate bezogen auf die Trainingsdaten. Im Umkehrschluss kann damit die Vermutung untermauert werden, dass die guten Generalisierungsleistungen des visuellen Systems auf den Realdaten COIL20 und COILselect daher rühren, dass auch reale Objekte in gewissem Umfang einen hierarchischen Aufbau bezüglich ihrer optischen Merkmale aufweisen.

Es stellt sich die Frage, warum das evolutionäre Entwurfsverfahren in 60% der Fälle keine Minima mit einer geringen Fehlklassifikation auf den Testdaten erreicht. Ein Grund hierfür kann in der Verkopplung der einfachen und der komplexen Merkmale liegen. Diese erschwert die evolutionäre Suche im Vergleich mit den bisherigen Entwurfsprozessen weiter. Ein weiterer Grund liegt in der Tatsache, dass bei den erzeugten Mustern die „Übergänge" von einer zur nächsten Ansicht nicht kontinuierlich, sondern diskret sind.

Betrachtet man die unterschiedlichen Systeme, die von der evolutionären Optimierung gefunden wurden und die alle eine 100%-ige Generalisierungsleistung zeigen, so ist – wie schon in Abschnitt 3.7 – eine große Variabilität festzustellen. Trotz der Einschränkungen des Systems enthält der Suchraum immer noch eine anscheinend große Menge von optimalen Lösungen, die eine 100%-ige Generalisierungsleistung aufweisen, bereit. In weiteren Untersuchungen könnten die Freiheitsgrade des Systems noch weiter reduziert werden bis zu der Grenze, an der nur noch **eine** vollständige Lösung des Problems möglich ist. Damit würde sich aber auch u.U. die evolutionäre Optimierung so weit vereinfachen, dass eine sinnvolle Korrespondenz zu dem auf Realdaten arbeitenden System nicht mehr bestehen würde.

Zusammenfassung

Zusammenfassend kann festgestellt werden, dass die Optimierung des hierarchischen visuellen Systems nur in 40% der Fälle kleine Trainingsfehler erreicht. In diesen Fällen aber wurde auch in gleichem Maße der Testfehler klein. Es kam zu keinem Overfitting der Trainingsdaten. Das bedeutet, dass das visuelle System über gute Generalisierungseigenschaften verfügt. Diese positive Eigenschaft ist wohl in hohem Maße auf den hierarchischen Aufbau des visuellen Systems zurückzuführen. Diese ist in idealer Weise dazu geeignet, eine Erkennungsaufgabe auf einer Musterdatenbank zu lösen, die selbst hierarchisch aufgebaut ist. Wenn man davon ausgeht, dass auch die Objekte der realen Welt in Bezug auf optische Merkmale über einen hierarchischen Aufbau verfügen, so ist anzunehmen, dass der Aufbau des visuellen Systems in besonderer Weise dazu geeignet ist, eine hohe Generalisierungsleistung bei entsprechenden Objekterkennungsaufgaben zu erbringen.

Kapitel 5

Kopplung des evolutionären Entwurfsverfahrens mit lokalem Lernen

In diesem Kapitel wird ein alternativer Optimierungsansatz des visuellen Systems dargestellt (vgl. auch [35, 36, 37]). Bei dem in Kapitel 3 dargestellten evolutionären Optimierungsansatz werden alle zu optimierenden Eigenschaften der visuellen Hierarchie direkt im Genom kodiert. So werden die Parameter der Systemnichtlinearitäten sowie jeder einzelne Eintrag in den Kombinationsmerkmalen als Fließkommazahlen kodiert. Durch diese direkte Form der Genotyp-Phänotyp-Abbildung sind die Evolutionsstrategien gezwungen, auf einem sehr hochdimensionalen Suchraum zu arbeiten. Zwar wurde in Kapitel 3 gezeigt, dass diese Suche erfolgreich ist, aber biologisch plausibel ist diese Form der Genotyp-Phänotyp-Abbildung nicht. Biologisch viel wahrscheinlicher ist die Form der indirekten Kodierung (vgl. Abschnitt 2.3).

Die Idee ist also, den Aufbauprozess jedes Phänotyps mit der Hilfe eines lokalen Lernvorgangs zu steuern. In dem in diesem Kapitel vorgeschlagenen Verfahren soll ein Teil der phänotypischen Eigenschaften des visuellen Systems weiterhin direkt in das Genom kodiert werden, wohingegen jedoch der überwiegende Teil durch ein unüberwachtes Lernverfahren bestimmt wird. Das Lernverfahren selbst kann über Parameter verfügen, die jetzt zusätzlich direkt in das Genom kodiert werden. Damit werden diese Steuerungsparameter auch Gegenstand der evolutionären Optimierung. Es existieren zwar eine Reihe von Arbeiten, die sich mit einer Kopplung von evolutionärer Optimierung und Lernen beschäftigen (vgl. auch Abschnitt 2.3), diese wurden bisher jedoch nicht in einer integrierten Form für den Entwurf von biologisch motivierten Objekterkennern eingesetzt. Das verwendete Lernverfahren operiert darüber hinaus auf Eingangsdaten, die durch die Umwelt mitbestimmt werden. Dies bedeutet in unserem Fall eine Mitbestimmung durch die Bilddaten, mit denen das visuelle System konfrontiert wird. Zum anderen werden die Eingangsdaten auch durch

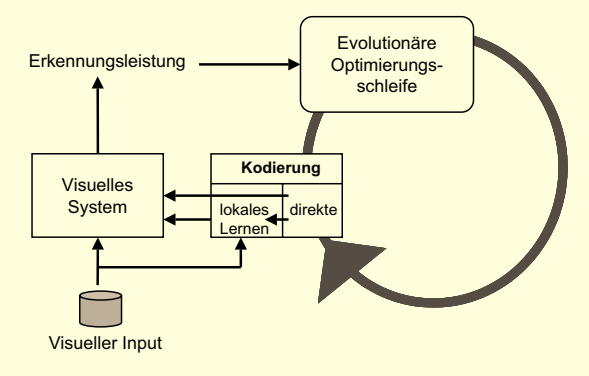

Abbildung 5.1: Schematische Darstellung der Optimierung des visuellen Systems mit Hilfe der Kopplung von Evolution und lokalem Lernen. Diese Form der Kodierung, bestehend aus einem direkten Anteil und einem lokalen Lernverfahren, wird im Folgenden als indirekte Kodierung bezeichnet.

die direkt kodierten Phänotypeigenschaften bestimmt. Die Lernalgorithmen erhalten keinen Sollausgang, sondern arbeiten unüberwacht. Das Ergebnis des Lernverfahrens bestimmt zusammen mit weiteren direkt kodierten Parametern den Aufbau des visuellen Systems. Auf diese Weise ist also ein unüberwachtes lokales Lernen in den Vorgang der evolutionären Optimierung fest eingebunden und mit diesem eng verzahnt. Eine schematische Darstellung findet sich in Abbildung 5.1.

Für das zur Verwendung kommende Lernen werden unterschiedliche Verfahren miteinander verglichen. Beim Aufbau der Repräsentation durch das Lernen wird das Prinzip der Spärlichkeit der Kodierung verfolgt, da es einerseits hierfür auch in biologischen Systemen Hinweise gibt (vgl. [2, 26, 3]) und andererseits auch die Leistungsfähigkeit eines solchen Ansatzes bei dem vorliegenden visuellen System schon nachgewiesen wurde [49, 50]. In der vorliegenden Arbeit soll die evolutionäre Optimierung mit einem lokalen Lernverfahren gekoppelt werden, um ein visuelles System zu optimieren, das in der Lage ist, eine Erkennung von dreidimensionalen Objekten durchzuführen. Im folgenden Abschnitt wird auf die entsprechend veränderte Kodierung des visuellen Systems eingegangen.

5.1 Kodierung durch Integration unüberwachter Lernverfahren

Aufgrund der Kopplung von Evolution und lokalem Lernen verändert sich die unter Abschnitt 3.2 dargestellte Kodierung des visuellen Systems folgendermaßen: Die direkte Kodierung der Kombinationsmerkmale entfällt zugunsten eines unüberwachten Lernverfahrens, das den Aufbau der Merkmale steuert.

Die Nichtlinearitäten des visuellen Systems $\gamma_1, \gamma_2, \theta_1, \theta_2, \sigma_1, \sigma_2$ werden weiterhin, wie bereits in Abschnitt 3.2 dargelegt, direkt kodiert. Zusätzlich werden etwaige Parameter, die den verwendeten lokalen Lernalgorithmus steuern, auch direkt in das Genom kodiert und so innerhalb der Evolution optimiert. Schematisch veranschaulicht ist die Kodierung in Abbildung 5.1.

Im Folgenden werden drei unterschiedliche Lernalgorithmen für die Einbettung in die evolutionäre Kodierung betrachtet. Diese sind die Hauptachsentransformation oder *Principle-Component-Analysis* (PCA), die schnelle unabhängige Hauptachsentransformation oder *fast-Independent-Component-Analysis* (fastICA) und das Verfahren der *non-negative-Sparse-Coding* (nnSC). Die Verfahren und ihre Verwendung werden im folgenden Abschnitt erläutert. Nur bei der Methode der non-negative-Sparse-Coding ist ein Steuerparameter notwendig, der zusätzlich zu den Systemnichtlinearitäten direkt in das Genom kodiert wird. Bei den Verfahren der PCA und fastICA sind in den verwendeten Implementierungen keine Steuerparameter notwendig, die optimiert werden müssten.

Unüberwachte Lernverfahren

Die zum Einsatz kommenden unüberwachten Lernverfahren sind zur Generierung der Kombinationsmerkmale bestimmt. Die Kombinationsmerkmale wiederum stellen die Verbindung der C1-Schicht mit der S2-Schicht dar. Durch diese Einbettung werden die Eingangsdaten der Lernalgorithmen aus der C1-Schicht entnommen. Und zwar werden dazu an zufälligen räumlichen Positionen[1] einer C1-Aktivierung 3×3-Pixel große Bildpatches ausgeschnitten. Da die C1-Schicht aus vier Ebenen besteht, erhält man auf diese Art je Bildpatch einen Eingangsvektor $\bar{\mathbf{c}}_1^{(p)}$ der Dimension 36 ($3 \times 3 \times 4 = 36$). In dieser Schreibweise stellt p den Index des Bildpatches dar. Eine schematische Darstellung der Extraktion der $\bar{\mathbf{c}}_1^{(p)}$ Vektoren ist in Abbildung 5.2 gegeben. Die p zufällig extrahierten Vektoren sind die Eingangsdaten für das lokale Lernverfahren, das die Kombinationsmerkmale generiert.

Die notwendigen Aktivierungen der C1-Schicht werden dadurch gewonnen, dass jedes der 1440 Bilder der COIL20 Datenbank bis zur C1-Schicht durch die visuelle Hierarchie propagiert wird. Dann wird aus je einer Aktivierung ein Vektor $\bar{\mathbf{c}}_1^{(p)}$ extrahiert. Auf diese Weise erhält man insgesamt 1440 Vektoren als Eingangsdaten für die unüberwachten lokalen Lernverfahren. Dieser Vorgang wird bei der Dekodierung jedes einzelnen Individuums in den entsprechenden Phänotyp durchgeführt. Die notwendigen Nichtlinearitäten der Schichten S1 und C1 stammen aus dem direkt kodierten Teil des entsprechenden Genoms. Um bei dem späteren Fitnessvergleich der einzelnen Individuen den Einfluss der zufälligen Bildpatchauswahl zu eliminieren, wird immer dieselbe räumliche Zufallsauswahl verwendet. Die $\bar{\mathbf{c}}_1^{(p)}$ Vektoren variieren jedoch von Individuum

[1] Diese zufällige räumliche Position ist jeweils identisch für die vier Ebenen der C1-Schicht.

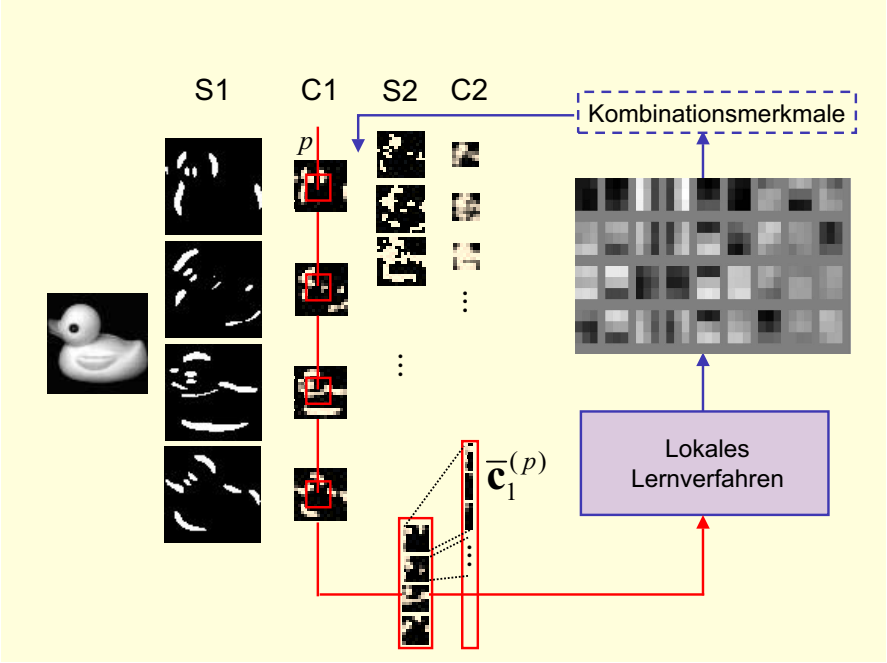

Abbildung 5.2: Schematische Darstellung der Extraktion der $\bar{\mathbf{c}}_1^{(p)}$ Vektoren. An einer zufälligen räumlichen Position einer C1-Aktivierung werden für alle vier Ebenen 3×3-Pixel große Bildpatches ausgeschnitten. Die Werte der Patches werden dann in den $\bar{\mathbf{c}}_1^{(p)}$ Vektor eingeordnet. Dieser Vorgang wird p-mal (einmal für jede C1-Aktivierung) durchgeführt. Die extrahierten Vektoren sind die Eingangsdaten für das lokale Lernverfahren, das die Kombinationsmerkmale generiert.

zu Individuum durch die unterschiedlichen Systemnichtlinearitäten, die die C1-Aktivierungen bestimmen. Damit handelt es sich bei der hier vorgeschlagenen Kopplung von Evolution und Lernen um eine Einbettung des Lernens mitten in den Dekodierungsvorgang der Individuen. Dies ist vergleichbar mit der biologischen Ontogenese.

Ein angewendetes Lernverfahren ist die PCA, die auf den 1440 36-dimensionalen $\bar{\mathbf{c}}_1^{(p)}$ Vektoren ausgeführt wird. Die damit ermittelten 36 Hauptachsen (der Dimension 36) können als Kombinationsmerkmale verwendet werden. Je nachdem wie hoch die Zahl L der verwendeten Kombinationsmerkmale sein soll, wird eine Unterauswahl in Abhängigkeit der Höhe der Eigenwerte getroffen. Und zwar werden die Hauptachsen mit den größten Eigenwerten ausgewählt. Die maximale Anzahl der möglichen Kombinationsmerkmale ist damit allerdings auf $L = 36$ beschränkt.

Ein der PCA ähnliches, aber im Allgemeinen leistungsfähigeres Verfahren, ist die unabhängige Hauptachsentransformation oder Independent-Component-Analysis (ICA). Im Unterschied zu der PCA werden die Hauptachsen als die Richtungen maximaler statistischer Unabhängigkeit gesucht, ohne die Randbedingung, dass diese senkrecht aufeinander stehen müssen. Die Bestimmung der einzelnen Hauptachsen ist jedoch aufwendiger und zeitraubender. Ein schneller Algorithmus wurde von Hyvärinen und Oja [20] vorgeschlagen. Dieser wird auch hier zur Berechnung der ICA eingesetzt.

Die dritte Methode, die zum Lernen der Kombinationsmerkmale eingesetzt werden soll, ist das Verfahren der non-negative-Sparse-Coding (nnSC) von Hoyer und Hyvärinen [17]. Anders als bei der ICA kann bei dieser Methode mit einer überbestimmten Basis gearbeitet werden. Damit fällt die Limitierung auf $L = 36$ Kombinationsmerkmale weg. Um weiterhin die Spärlichkeit der Basis zu erzwingen, wird über einen Spärlichkeitsfaktor (Sparsity) ein Term in eine Energiegleichung eingebracht, der eine gleichzeitige Benutzung von vielen Basisvektoren zur Rekonstruktion eines Eingangsvektors bestraft. Dazu wird bei dieser Lernregel die folgende Energiefunktion minimiert:

$$E = \sum_p ||\bar{\mathbf{c}}_1^{(p)} - \sum_l s_l^{(p)} \bar{\mathbf{w}}_2^l||^2 + \lambda \sum_p \sum_l s_l^{(p)}, \tag{5.1}$$

mit $l = 1, ..., L$, $L =$ Anzahl der Kombinationsmerkmale, mit $p = 1, ..., 1440$ und $p =$ Index des Bildpatches, sowie $\bar{\mathbf{w}}_2^l, s_l^{(p)} \geq 0$. Der linke Teil (der rechten Seite) der Gleichung (5.1) misst hierbei den Rekonstruktionsfehler des Bildpatches $\bar{\mathbf{c}}_1^{(p)}$ durch die Kombination der nicht orthogonalen Basisvektoren $\bar{\mathbf{w}}_2^l$. Der rechte Teil (der rechten Seite) hingegen erzwingt die spärliche Aktivierung der Koeffizienten $s_l^{(p)}$. Der Parameter λ steuert hierbei die Stärke dieser Spärlichkeitsbedingung.

Innerhalb des Lernverfahrens werden sowohl die Gewichtsvektoren $\bar{\mathbf{w}}_2^l$, die zugleich die Kombinationsmerkmale sind, als auch die Koeffizienten $s_l^{(p)}$ bestimmt. Nach einer zufälligen Initialisierung von $\bar{\mathbf{w}}_2^l$ erfolgt die Minimierung

mit Hilfe eines zweistufigen Gradientenabstiegverfahrens [26, 17]: In der ersten Stufe werden lediglich die Koeffizienten $s_l^{(p)}$ optimiert, während die zufällig gewählten Gewichte $\bar{\mathbf{w}}_2^l$ fest sind. In der zweiten Stufe werden dann die $\bar{\mathbf{w}}_2^l$ optimiert bei festen $s_l^{(p)}$ Werten. Beide Stufen werden bis zur Konvergenz der Parameter betrieben. Die Werte von $\bar{\mathbf{w}}_2^l$ und $s_l^{(p)}$ sind bei diesem Verfahren auf Werte größer gleich Null beschränkt. Da E nach unten beschränkt ist und für große Werte von $\bar{\mathbf{w}}_2^l, s_l^{(p)}$ nach unendlich strebt, konvergiert der Gradientenabstieg immer in ein lokales Minimum von E. Obwohl eine Reihe von lokalen Minima existieren, sind diese im Allgemeinen von ähnlicher Leistungsfähigkeit bezogen auf die Verwendung der damit gefundenen Kombinationsmerkmale (vgl. hierzu auch [3, 17, 49]).

Im Folgenden wird stichpunktartig nochmals die Kopplung von Evolution und Lernen dargestellt. Beschrieben wird das Dekodieren des Genotyps über die Erzeugung des Phänotyps bis hin zur Evaluation desselben. Die Parameter, die innerhalb dieses Prozesses evolutionär optimiert werden, sind fett gedruckt dargestellt.
Für jeden Genotyp der Population gilt:

1. Konstruktion des Phänotyps bis hin zur C1-Schicht mit Hilfe eines Teils der **Nichtlinearitäten**.

2. Generierung der C1-Schichtaktivierungen unter Benutzung der Datenbank A[2].

3. Ausschneiden und Sammeln der $4 \times 3 \times 3$-Bildpatches aus den C1-Schichtaktivierungen und Bildung der Vektoren $\bar{\mathbf{c}}_1^{(p)}$.

4. Benutzung des unüberwachten lokalen Lernens zur Generierung der Kombinationsmerkmale, unter Benutzung der $\bar{\mathbf{c}}_1^{(p)}$ Vektoren und (im Falle des non-negative-Sparse-Coding) des **sparsity**-Parameters.

5. Konstruktion des vollständigen Phänotyps – das visuelle System – mit allen **Nichtlinearitäten** und den Kombinationsmerkmalen.

6. Training des visuellen Systems mit den Trainingsansichten der Datenbank A (Abspeichern der C2-Aktivierungen mit den Labelinformationen).

7. Berechnung der Klassifikationsrate unter Benutzung der Testansichten der Datenbank A mit Hilfe des Nächsten-Nachbar-Klassifikators basierend auf den C2-Aktivierungen. Das Ergebnis ist die Fitness des Individuums.

Aufbauend auf dieser Kodierung und der damit verbundenen Kopplung von Evolution und lokalem Lernen wird jetzt die Optimierung des visuellen Systems durchgeführt.

[2] Im speziellen Fall wurde für die Datenbank A die COIL20 verwendet.

		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
PCA	9	9.4	10.6	1.1	23.4	25.1	1.2
	36	**8.1**	9.6	1.1	23.6	25.8	3.1
fast ICA	9	8.5	**9.4**	0.7	24.4	26.7	1.8
	36	8.8	9.7	1.0	22.5	24.7	2.8
nnSC	9	9.0	10.0	0.6	24.1	26.5	1.6
	36	8.5	9.7	1.3	22.4	**24.1**	1.4
	50	8.8	9.5	0.6	**21.7**	24.2	1.4

Tabelle 5.1: Ergebnisse der Optimierungen mit indirekter Kodierung. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. nnSC=non-negative-Sparse-Coding.

5.2 Aufbau und Ablauf des Entwurfsverfahrens

Die jetzt folgende evolutionäre Optimierung wird, so weit es möglich ist, mit denselben Optimierungsparametern wie in Kapitel 3 betrieben. Auf diese Weise wird der anschließende Vergleich der unterschiedlichen Optimierungen ermöglicht, d.h. es wird im Folgenden über 400 Generationen eine $(7, 19)$-Strategie verwandet. Der Strategieparameter σ_{km} der Kombinationsmerkmale entfällt, da der Aufbau der entsprechenden Merkmalsbank jetzt über die eingebetteten unüberwachten lokalen Lernverfahren durchgeführt wird. Neben der Anzahl L der Kombinationsmerkmale wird auch das verwendete lokale Lernverfahren in den Optimierungsläufen variiert. Wie zuvor, werden jeweils 10 Läufe mit unterschiedlichen Anfangsinitialisierungen durchgeführt. Für L werden die Werte 9, 36 und 50 verwendet. Für die Lernverfahren PCA und fastICA können allerdings keine Optimierungen mit $L = 50$ Merkmalen vorgenommen werden, da aufgrund der Dimensionalität der Eingangsdaten für diese Verfahren $L = 36$ $(4 \times 3 \times 3 = 36)$ die maximale Anzahl von Merkmalen darstellt.

5.3 Ergebnisse der Optimierung

Die erzielten Optimierungsergebnisse (Fehlklassifikationsraten) sind in Tabelle 5.1 dargestellt. Hierbei ist zu bemerken, dass das spezielle Verfahren der non-negative-Sparse-Coding (nnSC) die Einträge der hiermit bestimmten Kombinationsmerkmale auf positive Werte beschränkt. Das bedeutet, es können nur „positive" Kombinationsmerkmale gebildet werden. Bei der PCA und der fastICA hingegen können sowohl negative als auch positive Einträge in den Kom-

binationsmerkmalen genutzt werden. Aufgeführt sind die prozentualen Fehlklassifikationsraten auf der COIL20 und der COILselect Datenbank, jeweils mit dem besten und dem mittleren Ergebnis, sowie der Standardabweichung der jeweils 10 Optimierungsläufe.

Vergleicht man die Erkennungsleistungsunterschiede in Bezug auf die Generalisierung 1. Ordnung, die sich aufgrund der unterschiedlichen lokalen Lernverfahren ergeben, so kann man feststellen, dass sich das beste mittlere Ergebnis bei der fastICA bei $L = 9$ Merkmalen einstellt. Das beste Einzelergebnis wird bei der PCA mit $L = 36$ Merkmalen erzielt. Aufgrund der geringen Unterschiede und der relativ hohen Standardabweichungen sind jedoch die Unterschiede statistisch gesehen nicht signifikant.

In Bezug auf die Erkennungsleistung bei der Generalisierung 2. Ordnung ist festzustellen, dass das Verfahren der non-negative-Sparse-Coding (nnSC) insgesamt am leistungsfähigsten ist. Die bessere Performanz ist insofern noch höher zu bewerten da, wie schon erwähnt, hierbei nur positive Merkmalswerte erlaubt sind, welche – wie schon in Kapitel 3 nachgewiesen wurde – von geringerer Leistungsfähigkeit sind. Auf der anderen Seite verfügt die non-negative-Sparse-Coding Implementierung über den Vorteil, dass ein weiterer freier Parameter, die Sparsity, variabel an das visuelle Erkennungsproblem angepasst werden kann. Abschließend kann bemerkt werden, dass die Ergebnisse aller lokalen Lernverfahren relativ ähnlich sind und die Vorteile für das eine oder das andere Verfahren statistisch gesehen nicht signifikant sind.

Ein guter Wert für die Anzahl der Kombinationsmerkmale scheint wieder $L = 36$ zu sein. Genau wie in der direkten Optimierung bedeutet eine größere Anzahl von Merkmalen (ab einer gewissen minimalen Anzahl) keine Verbesserung der Leistungsfähigkeit des Systems mehr.

Die Ergebnisse der Generalisierung 2. Ordnung, ermittelt durch die Klassifikationsergebnisse auf der ORL-Gesichtsdatenbank, sind analog zu Tabelle 5.1 in Tabelle 5.2 dargestellt. Wie schon bei der COILselect Datenbank stellt sich heraus, dass das Verfahren non-negative-Sparse-Coding am leistungsfähigsten ist und das auch wieder bei einem Wert von $L = 36$ Kombinationsmerkmalen. Allerdings sind die Standardabweichungen der Ergebnisse größer, was mit der Unterschiedlichkeit der Bilddomänen zu erklären ist. So ist eine höhere Übertragbarkeit der Ergebnisse der Optimierung von der COIL20 zur COILselect Datenbank zu verzeichnen als von der COIL20 zu der Gesichtsdatenbank ORL. Aufgrund der hohen Standardabweichungen sind auch hier die Unterschiede der einzelnen indirekten Verfahren in Bezug auf ihre Fehlklassifikationsraten nicht signifikant.

In Abbildung 5.3 sind die Ergebnisse der Nichtlinearitäten des Szenarios mit $L = 50$ Kombinationsmerkmalen und non-negative-Sparse-Coding dargestellt. Die Darstellung ist analog zu Abbildung 3.5 auf Seite 62 mit dem Unterschied, dass jetzt ein weiterer Parameter, der Sparsity-Faktor, dargestellt ist. Wie auch schon im Falle der direkten Kodierung kann man feststellen, dass die ge-

		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler ORL [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
PCA	9	9.4	10.6	1.1	16.3	20.8	2.7
	36	**8.1**	9.6	1.1	**15.3**	22.7	12.5
fast ICA	9	8.5	**9.4**	0.7	18.8	23.8	3.6
	36	8.8	9.7	1.0	16.9	21.3	3.8
nnSC	9	9.0	10.0	0.6	19.7	28.8	9.1
	36	8.5	9.7	1.3	15.9	**19.8**	2.5
	50	8.8	9.5	0.6	16.3	21.7	5.1

Tabelle 5.2: Ergebnisse der Optimierungen mit indirekter Kodierung. Die Generalisierung 2. Ordnung wurde hier mit Hilfe der ORL-Gesichtsdatenbank ermittelt. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe. nnSC=non-negative-Sparse-Coding.

fundenen Sätze der Nichtlinearitäten recht unterschiedlich sind. Ebenso der Sparsity-Faktor streut relativ stark in einem Intervall von 0.1 bis 2.0. Auch hier sieht es so aus, dass sich in Bezug auf die beiden Werte γ_1 und θ_1 die beiden unterschiedlichen Strategien herausbilden: Nämlich entweder eine hohe Selektivität durch einen hohen Winner-Take-Most-Parameter gefolgt von einer geringen Selektivität durch einen geringen Schwellwert θ_1 oder genau das Umgekehrte. Nur hier im Falle der Kopplung von Evolution und Lernen stellt sich die umgekehrte Tendenz in Bezug auf die Generalisierung heraus. Jetzt zeigt die Strategie mit dem erhöhten Winner-Take-Most-Parameter eine leicht bessere Generalisierung 1. Ordnung.

Wie im Falle der direkten evolutionären Optimierung werden nun die durch die Evolution gefundenen Kombinationsmerkmale miteinander verglichen. Zwei optimierte Kombinationsmerkmalsbänke sind in Abbildung 5.4 dargestellt. Diese weisen, anders als im Falle der direkten Optimierung, schon augenscheinlich eine größere Ähnlichkeit zueinander auf. Dieser Eindruck wird auch durch die nachfolgende quantitative Analyse mit Hilfe des Distanzmaßes (siehe Gleichung (3.1) auf Seite 65) bestätigt. Hierzu werden die Abstände aller möglichen Paarungen der besten Bänke (Endergebnis je eines der 10 Optimierungsläufe) für den Fall: $L = 9$ Kombinationsmerkmale (KM) und nnSC berechnet. Um die Distanzwerte einordnen zu können, werden zunächst 10 Merkmalsbänke mit gleichverteilten Zufallszahlen im Intervall [0,1] erzeugt. Hierbei ergibt sich eine mittlere Distanz von $\bar{D}_{KM} = 20.2$, bei einem geringsten Abstand von $D_{KM}^{min} = 19.2$ und einer Standardabweichung von $D_{KM}^{std} = 0.5$. Um zu erfahren, wie sensitiv sich der Abstand zu der Klassifikationsleistung verhält, wurde zu den Kombinationsmerkmalsbänken der beiden besten Optimierungs-

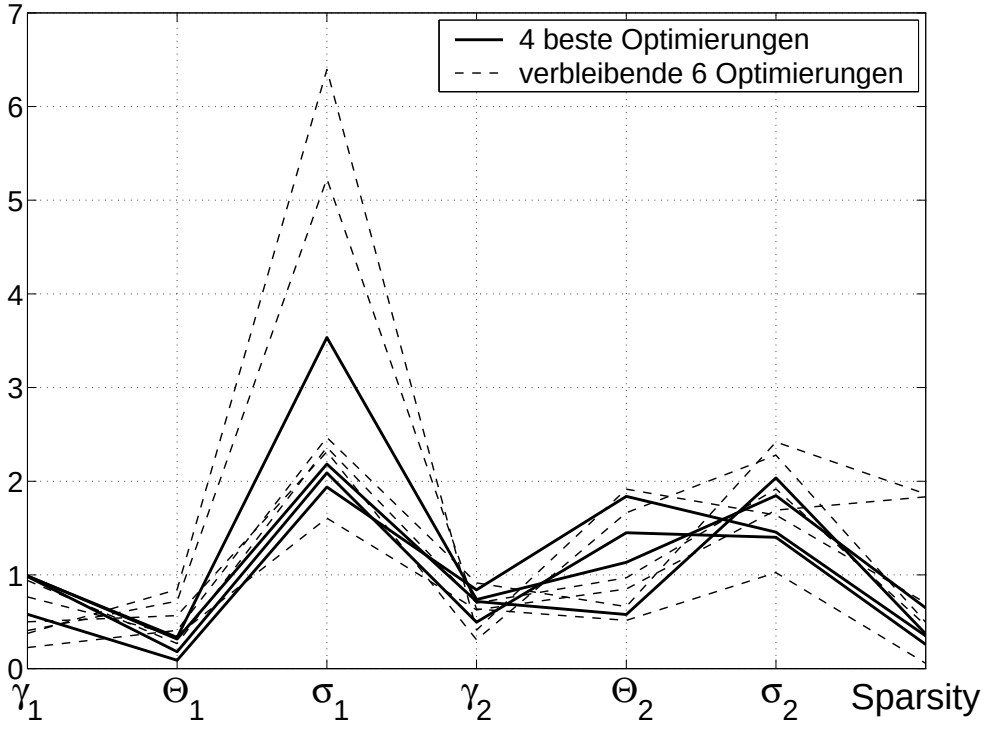

Abbildung 5.3: Werte der Nichtlinearitäten des jeweils besten Individuums eines der 10 Optimierungsläufe bei $L = 50$ Kombinationsmerkmalen im Falle der non-negative-Sparse-Coding.

L=9 nnSC		KM+N(0,0.025)			KM+N(0,0.05)			KM+N(0,0.1)			zufällige KM		
		b	m	s	b	m	s	b	m	s	b	m	s
Bester Lauf (8.8 %)	D	1.3	1.4	0.07	2.4	2.6	0.1	4.5	4.8	0.2	23.1	24.4	0.8
	Fehler COIL20 [%]	10.4	11.0	0.8	11.0	12.1	0.9	12.5	14.2	1.1	15.2	17.5	2.0
Zweitbester Lauf (9.8 %)	D	1.3	1.4	0.07	2.3	2.6	0.1	4.7	5.1	0.2	22.9	24.0	0.8
	Fehler COIL20 [%]	10.6	12.2	0.9	12.1	13.7	1.1	13.1	16.1	2.0	15.4	17.3	1.1

Tabelle 5.3: Distanzen („D") und Klassifikationsfehler der beiden besten visuellen Systeme ($L = 9$ non-negative-Sparse-Coding (nnSC)) nach dem Verrauschen der Kombinationsmerkmale („KM") durch Addition normalverteilter Zufallszahlen unterschiedlicher Standardabweichung. Die Fehlklassifikationsraten der unverrauschten Systeme sind 8.8% und 9.8% (aufgeführt in Klammern). Es wurden je 10 Experimente durchgeführt, wobei „m" den mittleren, „s" die Standardabweichung und „b" den besten bzw. kleinsten Wert angibt.

läufe ($L = 9$ mit nnSC) normalverteiltes Rauschen addiert. Die verwendeten Standardabweichungen des Rauschens sind: $\sigma_{noise} = 0.025, 0.05, 0.1$. Dieser Vorgang wurde 10-mal für die drei verschiedenen Rauschstärken durchgeführt. Zuletzt wurde die Bank vollständig durch eine Zufallsbank ersetzt. Nach dem Stören der ursprünglichen Bank mit Rauschen wurde der Abstand zur Ursprungsbank und die neue Fehlklassifikationsrate des veränderten visuellen Systems berechnet. Das Ergebnis dieser Untersuchung ist in Tabelle 5.3 zu finden.

Es zeigt sich (noch stärker als im Falle der direkten Optimierung, vgl. Tabelle 3.4 Seite 67), dass der mittlere Klassifikationsfehler in beiden optimierten Erkennungssystemen schon durch die Zugabe von relativ geringem Rauschen stark ansteigt. Daraus kann geschlussfolgert werden, dass es sich bei den gefundenen Optima um relativ enge Bereiche in der Fitnesslandschaft handelt[3] (enger noch als im Falle der direkten Optimierung).

Die Analyse der Abstände der 10 optimierten Merkmalsbänke ergibt einen mittleren Abstand von $\bar{D}_{KM} = 5.2$ und einen minimalen Abstand von $D_{KM}^{min} = 1.9$, bei einer Standardabweichung von $D_{KM}^{std} = 1.4$. Vergleicht man diesen Abstand mit dem mittleren Abstand, der sich bei den Zufallsbänken ergeben hat ($\bar{D}_{KM}(\text{Zufallsbänke}) = 20.2$), so kann festgestellt werden, dass sich die 10 optimierten Kombinationsmerkmalsbänke, die jeweils zum besten visuellen System gehören, relativ ähnlich sind. D.h., anders als im Falle der direkten Optimierung haben hier nicht nur visuelle Systeme mit einem kleinen Abstandsmaß eine ähnliche Fitness, sondern auch Systeme mit ähnlicher Fitness ein kleines

[3] An dieser Stelle sei noch einmal bemerkt, dass das Rauschen auf die fertig optimierten Systeme angewendet wird. Das unüberwachte Lernen kommt nicht mehr zum Einsatz und kann damit auch nicht mehr das Optimum verbreitern.

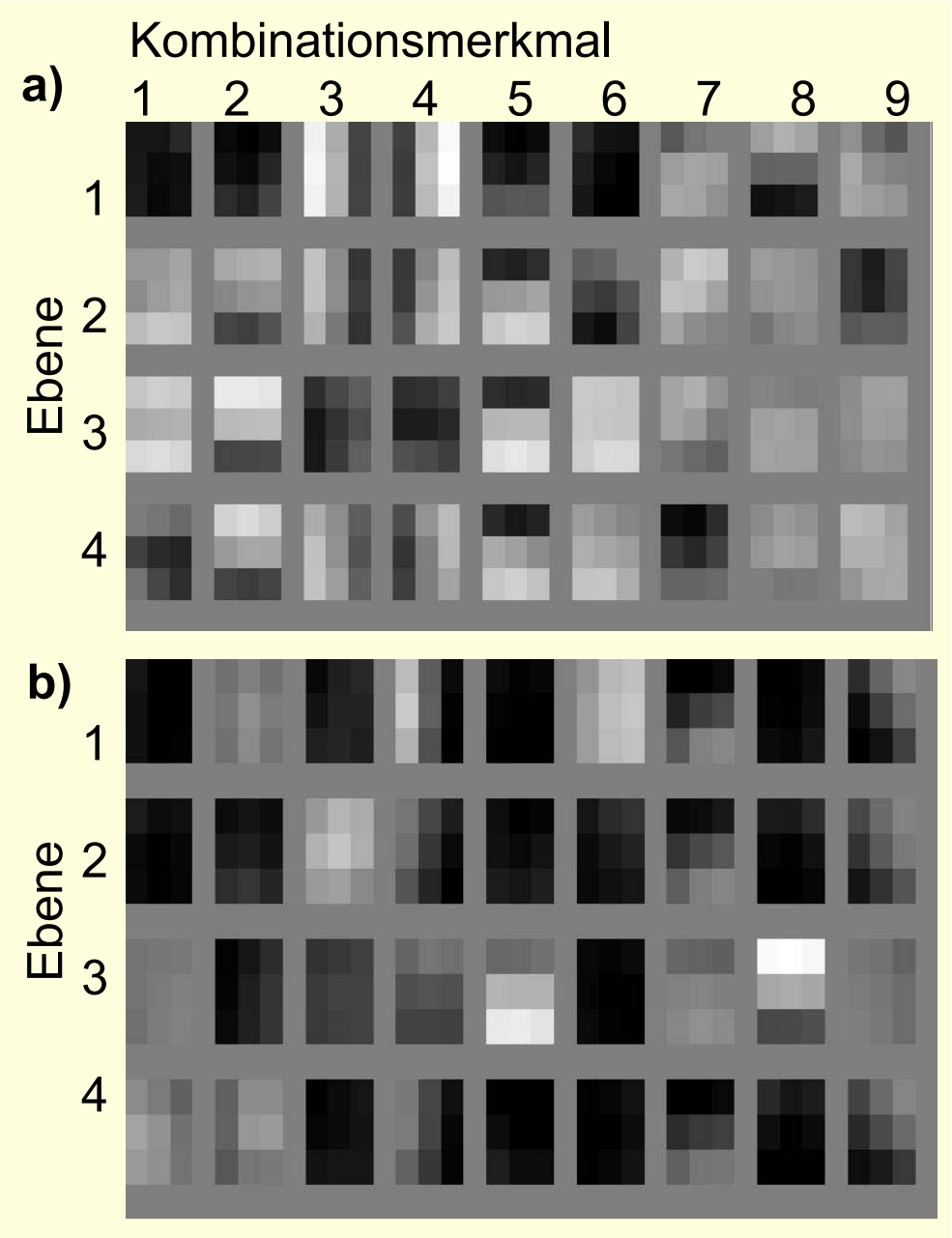

Abbildung 5.4: Optimierte Kombinationsmerkmalsbänke für den Fall $L = 9$ mit non-negative-Sparse-Coding (nnSC). Die Werte reichen von 0 (schwarz) bis $+1$ (weiß). In den Spalten stehen die unterschiedlichen Merkmale, in den Zeilen sind die jeweiligen Ebenen der Merkmale dargestellt. a) bester Optimierungslauf b) zweitbester Optimierungslauf.

Abstandsmaß.

Der Hauptgrund für die Ähnlichkeit der Kombinationsmerkmalsbänke liegt in der Tatsache, dass die Bänke im Falle der Kopplung von Evolution und Lernen in jedem Einzelfall von demselben unüberwachten Lernverfahren erzeugt werden. Der Input, der den Lernverfahren zur Verfügung gestellt wird, wird jedoch durch die unterschiedliche Wahl der Nichtlinearitäten beeinflusst. Ebenso werden im Falle des non-negative-Sparse-Coding auch unterschiedliche Sparsity-Faktoren verwendet.

Im Folgenden soll jetzt der Zeitverlauf der Optimierung betrachtet werden. Hierzu wird exemplarisch der beste Optimierungslauf für $L = 50$ und nnSC, dargestellt in Abbildung 5.5, untersucht.

Dargestellt sind die Fehlklassifikationsrate, der Strategieparameter und die Nichtlinearitäten während der gesamten Optimierung. Gleichzeitig gezeigt werden die sieben besten Individuen einer Generation. Auf diese Weise wird auch die Variationsbreite der Elternpopulation, die ja auf sieben festgelegt ist, veranschaulicht. Festzustellen ist auch hier (im Vergleich zu der direkten Optimierung) die typische Abnahme der Mutationsschrittweite der Nichtlinearitäten σ_{nl} (über den gesamten Verlauf betrachtet). In diesem Lauf ist jedoch zunächst ein Ansteigen der Schrittweite zu verzeichnen, das bedeutet, dass sich zu Beginn der Optimierung eine Phase erhöhter Exploration des Suchraumes als nützlich erwiesen hat. In Bezug auf die Fehlklassifikationsrate kann festgestellt werden, dass es zu Beginn der Optimierung zu einem schnellen, annähernd linearen Abfall des Fehlers kommt (s.a. Ausschnittvergrößerung). So fällt die Fehlklassifikationsrate in den ersten 65 Generationen auf einen Wert von 9.4% und erreicht in den folgenden 335 Generationen nur noch eine weitere Verbesserung um 0.6% auf einen Wert von 8.8%. Dieses Verhalten kann mit der Einführung des lokalen Lernens begründet werden, welches jetzt die Optimierung der Kombinationsmerkmale übernimmt. Damit ist die Optimierungsaufgabe, die jetzt einen 7-dimensionalen Fitnessraum umfasst, merklich vereinfacht worden.

Betrachtet man den Zeitverlauf der Nichtlinearitäten (einschließlich des Sparsity-Faktors) so kann beobachtet werden, dass der größere Teil der Adaptionsdynamik sich im ersten Viertel der Optimierung abspielt. Diese Entwicklung ist zum Teil durch den damit einhergehenden Abfall des Strategieparameters σ_{nl} bedingt.

5.4 Vergleich von Optimierung mit und ohne lokalem Lernen

In diesem Abschnitt werden die Ergebnisse der unterschiedlichen evolutionären Optimierungen (direkte und indirekte Kodierung) miteinander verglichen. In Tabelle 5.4 sind hierzu die Fehlklassifikationsraten der beiden Ansätze noch

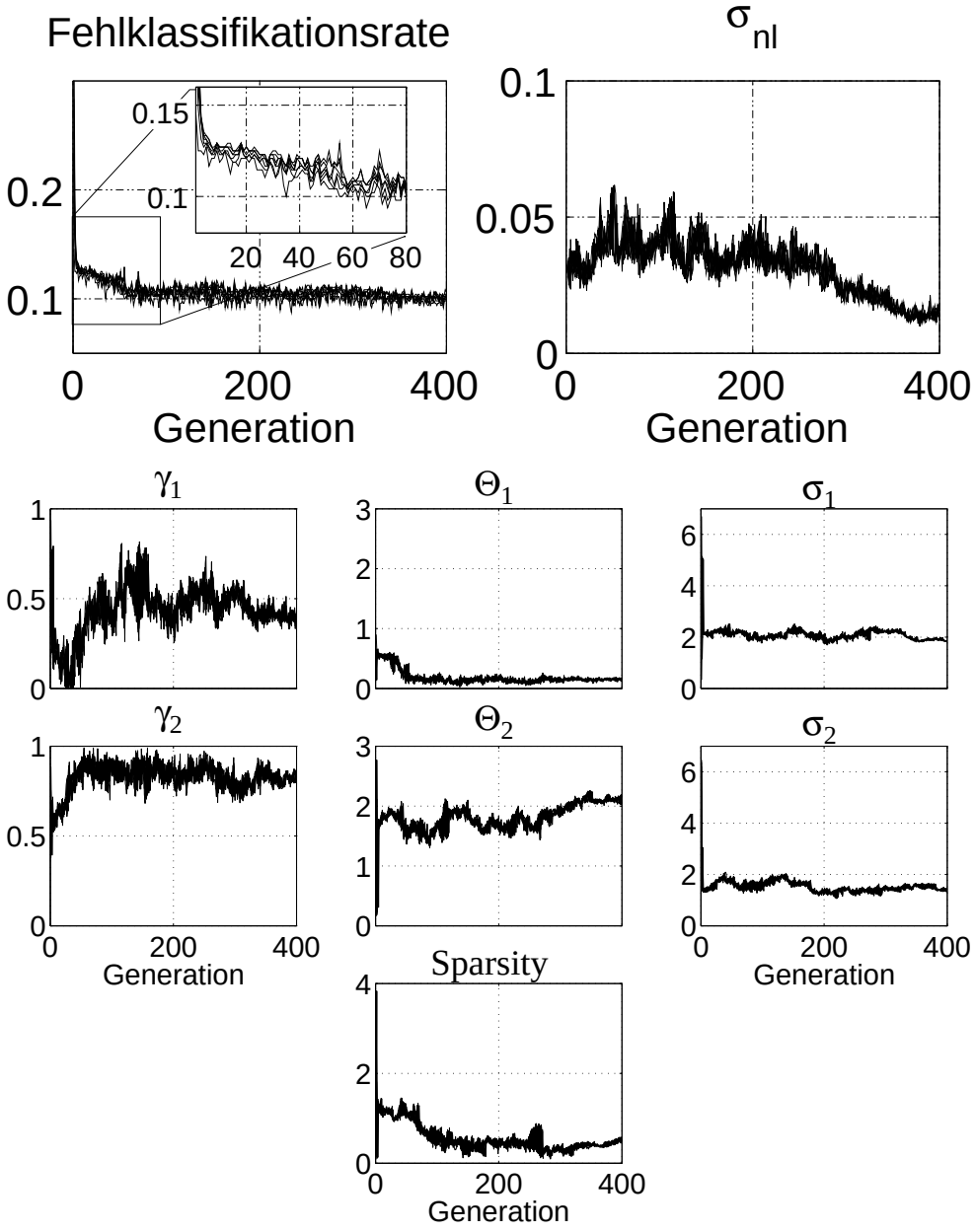

Abbildung 5.5: Zeitverlauf von Fehlklassifikationsrate (negative Fitness), Strategieparameter σ_{nl} und den Nichtlinearitäten der 7 besten Individuen während des besten Optimierungslaufes mit $L = 50$ und nnSC. Die 7 besten Individuen, die in der nächsten Generation zu Eltern werden, sind übereinander dargestellt und visualisieren auf diese Weise die Variationsbreite der Elternpopulation im Parameterraum.

direkte Kodierung		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
pos. KM	9	7.9	8.6	0.5	24.2	27.6	3.0
	36	7.7	8.3	0.5	23.2	26.5	2.2
	50	7.3	8.6	0.8	23.2	24.8	1.5
neg. KM	9	**6.5**	8.1	1.2	22.8	26.3	2.4
	36	7.1	**7.8**	0.5	22.9	24.2	1.8
	50	7.1	8.1	0.7	22.4	24.3	1.7

indirekte Kodierung		Fehler COIL20 [%] (Generalisierung 1. Ordnung)			Fehler COILselect [%] (Generalisierung 2. Ordnung)		
	L	b	m	s	b	m	s
PCA	9	9.4	10.6	1.1	23.4	25.1	1.2
	36	8.1	9.6	1.1	23.6	25.8	3.1
fast ICA	9	8.5	9.4	0.7	24.4	26.7	1.8
	36	8.8	9.7	1.0	22.5	24.7	2.8
nnSC	9	9.0	10.0	0.6	24.1	26.5	1.6
	36	8.5	9.7	1.3	22.4	**24.1**	1.4
	50	8.8	9.5	0.6	**21.7**	24.2	1.4

Tabelle 5.4: Ergebnisse der direkten (vgl. Tabelle 3.2) und der indirekten (vgl. Tabelle 5.1) evolutionären Optimierungen. L=Anzahl der Kombinationsmerkmale, b=bestes Ergebnis, m=mittleres Ergebnis und s=Standardabweichung der 10 Läufe.

einmal zusammengestellt.

Es wird deutlich, dass die direkte Kodierung auf der COIL20 Datenbank eine geringere Fehlklassifikationsrate erreicht, d.h., dass also die Generalisierung 1. Ordnung bei der direkten Kodierung besser ist. Dies gilt für jede Anzahl L der Kombinationsmerkmale. Vergleicht man die Ergebnisse mit gleicher Anzahl von Kombinationsmerkmalen, so ist die direkte Kodierung signifikant besser als die indirekte Kodierung. Die Signifikanzanalyse wurde mit Hilfe des Student-t-Tests durchgeführt.

Betrachtet man die Erkennungsleistung auf der COILselect Datenbank, d.h. die Generalisierung 2. Ordnung, so lässt sich feststellen, dass hier das bessere Ergebnis nicht durchgängig von einer Kodierung erreicht wird. Das beste mittlere Ergebnis, insgesamt gesehen, wird von der indirekten Kodierung mit non-negative-Sparse-Coding (nnSC) erreicht, ebenso wie das beste Einzelergebnis. Der kleine Performanzvorsprung ist allerdings nicht statistisch signifikant. Jedoch muss auch in Betracht gezogen werden, dass auf Grund des nnSC

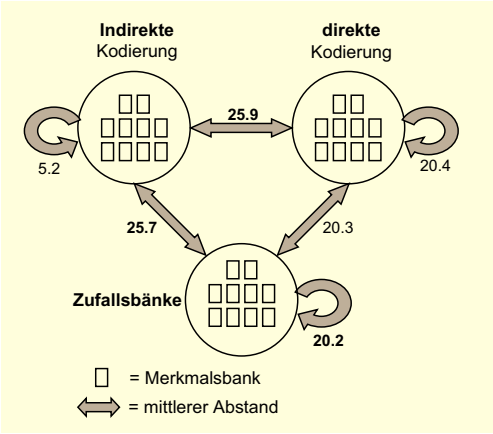

Abbildung 5.6: Darstellung der mittleren Abstände verschiedener Merkmalsbänke zueinander für den Fall von $L = 9$ positiven KM.

Verfahrens in diesem Fall keine Kombinationsmerkmale mit negativen Werten aufgebaut werden. Diese sind jedoch (wie der Vergleich der direkt kodierten Verfahren zeigt) leistungsfähiger.

Vergleicht man die unterschiedlichen Kombinationsmerkmalsbänke (siehe Abbildungen 3.7 und 5.4 auf den Seiten 64 und 110) der beiden unterschiedlichen Ansätze, so kann man rein optisch erkennen, dass diese sehr unterschiedlich in ihrem Aufbau sind. Die quantitative Untersuchung mit Hilfe des Distanzmaßes D_{KM} bestätigt diesen Eindruck. So beträgt der mittlere Abstand der 10 besten Ergebnisse (jeweils das beste Ergebnis aus einem der 10 unterschiedlichen Optimierungsläufe) zwischen indirekter (nnSC, $L = 9$ positive KM) und direkter Kodierung ($L = 9$ positive KM) $\bar{D}_{KM} = 25.9$. Zur Einordnung dieses Wertes sei bemerkt, dass der mittlere Abstand positiver Zufallsbänke zueinander $\bar{D}_{KM} = 20.2$ und der mittlere Abstand zwischen indirekt kodierten Bänken (nnSC, $L = 9$ positive KM) und Zufallsbänken $\bar{D}_{KM} = 25.7$ beträgt. Die mittleren Abstände zwischen den unterschiedlichen Merkmalsbänken sind graphisch noch einmal in Abbildung 5.6 dargestellt. Es ergibt sich damit, dass die Merkmalsbänke von direkter und indirekter Kodierung sehr unterschiedlich sind; unterschiedlicher als reine Zufallsbänke zueinander.

Dieser große Unterschied ist damit zu erklären, dass die Struktur der indirekt kodierten Kombinationsmerkmale vom Aufbau her dem der direkt kodierten KM und dem der Zufallsbänke sehr unähnlich ist. Im Gegensatz zu den direkt kodierten Bänken und den Zufallsbänken herrschen bei den indirekt kodierten KM gleichmäßige Übergänge zwischen den einzelnen Werten einer 3×3-Ebene.

Eine andere Möglichkeit, die Kombinationsmerkmalsbänke zu analysieren, ist die Anwendung der Hauptachsenanalyse (PCA) auf die C2-Schichtaktivierungen (also die Aktivierungen, die unmittelbar durch die Faltung mit den Kombinationsmerkmalen erzeugt werden). Auf diese Weise kann auch die Fra-

ge untersucht werden, wie stark die Abhängigkeiten der einzelnen Merkmale untereinander sind. Außerdem wird erkennbar, wie viele und wie stark die Merkmale, z.B. im Falle von $L = 50$ tatsächlich bei der Erkennung verwendet werden.

Hierzu wird eine Menge von 72000 Vektoren

$$\mathbf{x}_i = (c_2^1(x,y), c_2^2(x,y), ..., c_2^L(x,y))^\top \in I\!R^L$$

an 10 zufälligen Positionen (x, y) von C2-Schichtaktivierungen zusammengestellt, d.h. 10 zufällige Positionen für jede C2-Schichtaktivierung, die von je einem der 7200 COIL100 Bilder erzeugt wurde. Es sei noch einmal bemerkt, dass die Anzahl der unterschiedlichen Ebenen der C2-Schicht durch die Anzahl L Kombinationsmerkmale gegeben ist. Der Test wird bei dem jeweils besten visuellen System, bezogen auf die Generalisierung 2. Ordnung, für die Fälle $L = 9, 50$ durchgeführt. Verglichen werden die folgenden Systeme: positive Kombinationsmerkmale mit der direkten Kodierung („direkte Kodierung") und Kopplung von Evolution und Lernen mit dem Verfahren der non-negative-Sparse-Coding („indirekte Kodierung"). Die Ergebnisse sind in Abbildung 5.7 dargestellt.

Die normalisierten Hauptachsen wurden in der Abbildung mit Grauwerten dargestellt. Hierbei wurde schwarz für den kleinsten negativen Wert und weiß für den größten positiven Wert gewählt. Die Spalten in der Darstellung korrespondieren mit den einzelnen Hauptachsen. Unter jeder Hauptachse ist der entsprechende Eigenwert aufgetragen.

Im Falle von $L = 9$ sind die erhaltenen Ergebnisse von direkter und indirekter Kodierung sehr ähnlich: Alle L Hauptachsen haben ein ähnliches Spektrum an Eigenwerten. Außerdem sind die Hauptachsen selbst insofern ähnlich, als sie keines der L Merkmale ungenutzt lassen. Das kann man daran erkennen, dass alle Zeilen Werte ungleich Null beinhalten. Der Raum der C2-Aktivierungen wird also effizient durch die L Merkmale aufgespannt.

Es ist zu bemerken, dass die Hauptachse, bei der alle Komponenten mit einem gleich großen Anteil aktiviert sind (jeweils die Hauptachse am rechten Rand der Darstellung), einen Eigenwert von Null hat. Dieser Umstand ist durch die Winner-Take-Most-Nichtlinearität begründet, welche eine gleichzeitige gleich starke Aktivierung von unterschiedlichen Merkmalen verhindert.

Im Falle von $L = 50$ liegt jedoch ein anderer Sachverhalt vor. Die beiden Spektren der Eigenwerte sind in diesen Fällen unterschiedlich: Die Verteilung der direkten Kodierung ist weitaus steiler. Dies wird bei der Betrachtung der unterschiedlichen Skalierung der Hochachsen deutlich. Bei der Untersuchung der einzelnen Hauptachsen fällt zudem auf, dass viele Merkmale bei der direkten Kodierung gar nicht verwendet werden. Das ist an der Ausbildung der „horizontalen Banden" zu erkennen. Zwar werden die entsprechenden Merkmale von einzelnen Hauptachsen mit einbezogen, jedoch ist der Beitrag dieser Achsen sehr gering, da ihnen sehr kleine Eigenwerte zugeordnet sind.

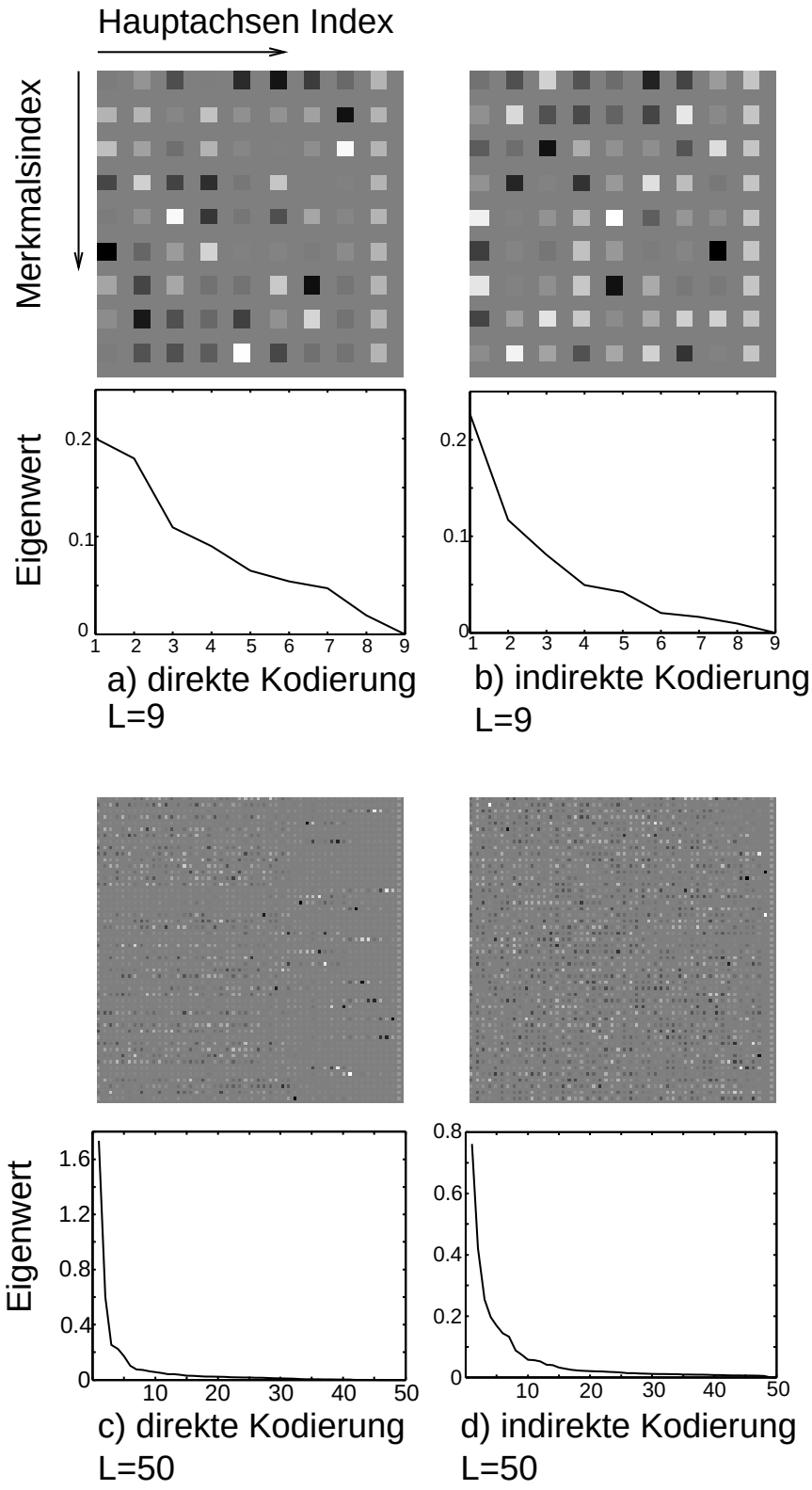

Abbildung 5.7: Hauptachsenanalyse auf dem Ausgang der Kombinationsmerkmale an einer bestimmten räumlichen Position in der C2-Schicht. (Bemerkt sei, dass durch die unterschiedliche Skalierung der y-Achsen in c) und d) die steilere Verteilung der Eigenwerte im Falle der direkten Kodierung besser illustriert wird.)

Das bedeutet, dass im Falle von $L = 50$ das visuelle System, das mit Hilfe der direkten Kodierung optimiert wurde, weniger Merkmale effektiv verwendet als das visuelle System, das mit Hilfe der indirekten Kodierung optimiert wurde. Der C2-Raum wird schlechter durch die 50 Hauptachsen aufgespannt. Der Grund dafür, dass die indirekte Kodierung die ihr zur Verfügung stehenden Merkmale besser ausnutzt, liegt in der Verwendung des lokalen Lernens. Dieses begünstigt die gleichmäßigere Verwendung der verschiedenen Merkmale.

5.5 Vergleich der Ergebnisse mit anderen Erkennungssystemen

Um die Leistungsfähigkeit des dargestellten evolutionären Optimierungsverfahrens besser einschätzen zu können, wird im Folgenden die Erkennungsleistung des besten optimierten Systems mit den Erkennungsleistungen anderer Ansätze verglichen. In den Vergleich eingeschlossen wird das zuvor manuell optimierte visuelle System [49], in der Tabelle mit „mVS" bezeichnet. Bei diesem System wurde bereits das Verfahren der unüberwachten spärlichen Kodierung (nnSC) zur Bestimmung der Kombinationsmerkmale verwendet, jedoch ohne Einbettung in die evolutionäre Entwurfsmethode. Geeignete Werte für die Nichtlinearitäten wurden in diesem System mit Hilfe einer Rastersuche ermittelt. In die Vergleichstabelle wurde auch die Erkennungsleistung eines reinen Nächsten-Nachbar-Erkenners mit aufgenommen, in der Tabelle mit „NNE" bezeichnet. Mit „Columbia" ist der Erkenner von Nayar et al. bezeichnet [25]. Dieser basiert auf einer Transformation der Bilder in den Raum ihrer Hauptachsen. Das Erkennungssystem von Roobaert und van Hulle [33], das die Methode der Support-Vector-Machine (SVM) verwendet, ist mit „SVM" bezeichnet. Bei den dargestellten Erkennungsleistungen wurden sowohl die Anzahl der zu erkennenden Objekte variiert als auch die Anzahl der verwendeten Trainingsansichten. In den Fällen, in denen weniger als 100 Objekte erkannt werden mussten, wurde die entsprechende Anzahl immer startend von dem 1. Objekt in der gegebenen Reihenfolge der COIL100 Objekte genommen. Die Trainingsansichten wurden wie zuvor so ausgewählt, dass ihre Ansichtswinkel einen maximalen Abstand haben. Der Test wurde immer auf allen übrigen Ansichten durchgeführt. Bei dem evolutionär optimierten visuellen System („optVS") handelt es sich um das mittels der Kopplung von Evolution und Lernen für $L = 50$ Kombinationsmerkmale optimierte System. Das verwendete unüberwachte Lernverfahren war das non-negative-Sparse-Coding. In allen Tests wurde das System eingesetzt, das innerhalb der Optimierung immer drei Trainingsansichten verwendete. Dieses System wurde jetzt mit vier Trainingsansichten belehrt. Um einen sinnvollen Vergleich mit dem manuell optimierten visuellen System zu haben, wurde genau wie dort die Klassifikation auf der letzten Schicht – im Gegensatz zu bisher – nicht von einem Nächsten-Nachbar-Klassifikator, sondern mit Hilfe

	30 Objekte			4 Trainingsansichten		
	Trainingsansichten			Anzahl der Objekte		
Methode	36	8	2	10	30	100
NNE	0	7.5	29.5	13.5	18.2	29.9
Columbia	0	4.4	32.9	7.9	15.4	23.0
SVM	0	4.8	29.0	9.0	15.1	25.4
mVS	0	7.3	28.3	18.4	15.8	23.9
optVS	0	4.4	22.9	12.4	12.9	20.2

Tabelle 5.5: Vergleich der Fehlklassifikationsraten unterschiedlicher Erkennungssysteme auf der COIL100 Datenbank.

eines Single-Layer-Perceptrons durchgeführt.

Der Vergleich der Ergebnisse ist in Tabelle 5.5 dargestellt. Von besonderem Interesse sind hierbei die dritte und die letzte Spalte, da in diesen die Ergebnisse für die komplizierteren Erkennungsaufgaben stehen: Die Erkennung von vielen Objekten nach einem Training mit wenigen Trainingsansichten. In dem Fall der Erkennung von 30 Objekten nach dem Training mit zwei Trainingsansichten konnte das manuell optimierte visuelle System um 5.4% und in dem Fall von 100 Objekten und vier Trainingsansichten um 3.7% verbessert werden. Damit ist die Erkennung mit Hilfe des biologisch inspirierten visuellen Systems auch bei diesen Aufgabenstellungen am besten. Nur bei der Erkennung von lediglich 10 Objekten nach einem Training mit vier Ansichten zeigt das visuelle System eine schlechtere Erkennungsleistung als der SVM Erkenner und das System von Nayar et al. („Columbia").

Eine interessante Arbeit, die sich mit der evolutionären Merkmalsextraktion auch im Zusammenhang mit der visuellen Objekterkennung am Beispiel der COIL Datenbank beschäftigt, ist die von Brown et al. [5]. Die hier vorgeschlagene Methode dient der Extraktion von visuellen Merkmalen aufbauend auf der Schicht der C1-Aktivierungen aus [49]. Nachdem diese Aktivierungen für alle Bilddaten generiert worden sind, werden diese in Trainings- und Testansichten aufgeteilt. Mit diesen Daten wird mit einem Backpropagation-Algorithmus ein Ensemble von MLPs auf die Lösung des Klassifikationsproblems trainiert. Diese werden jedoch nicht zur finalen Objektklassifikation benutzt, vielmehr werden aus diesen Netzen geeignete versteckte Neuronen, inklusive ihrer Gewichte, als Einzelmerkmale herausextrahiert. Hierbei werden jeweils die versteckten Neuronen selektiert, die einerseits einen besonders großen Beitrag zur Klassifikation leisten, und andererseits auch möglichst unterschiedlich zueinander sind. Um diese Unterschiedlichkeit sicherzustellen, wird das Verfahren des *Negative-Correlation-Learning* benutzt. Nachdem eine bestimmte Zahl von versteckten Neuronen auf diese Art extrahiert wurde, führt auf diesen dann ein Nächster-Nachbar-Klassifikator die Objekterkennung durch. Mit dieser Methode wird

auf den ersten 20 Objekten der COIL100 Datenbank, bei vier Trainingsansichten, ein Klassifikationsfehler von 14.6% erreicht. Die Erkennungsleistung des Systems von Brown et al. ist als eine Generalisierungsleistung 1. Ordnung zu betrachten, da bei der Extraktion der Merkmale aus den trainierten MLPs die Netze und Knoten gewählt wurden, die eine gute Klassifikationsleistung auf den Testdaten aufwiesen. Auf diese Weise waren die Testdaten an dem Entwurf des optimalen Erkenners beteiligt. Die Klassifikationsleistung von 14.6% liegt zwar nur 0.8% über der Nächsten-Nachbar-Klassifikation auf den C1-Aktivierungen (die Basis für die Merkmalsextraktion), jedoch konnte durch dieses Verfahren die Dimension des Merkmalsraumes von 1024 (Dimension einer C1-Aktivierung) auf 300 Dimensionen reduziert werden.

Im Vergleich dazu ist die Dimension des innerhalb dieser Arbeit optimierten visuellen Systems bei $L = 9$ Kombinationsmerkmalen 576. Jedoch beträgt der kleinste Klassifikationsfehler auf den 20 Objekten der COIL20 Datenbank 8.1% bei nur drei (im Gegensatz zu vier) Trainingsansichten[4]. Das entsprechende visuelle System wurde bei der evolutionären Optimierung mit indirekter Kodierung gefunden. Bei der direkten Kodierung wurde ein Fehler von 6.5% erreicht.

Eine weitere gute Möglichkeit, die domänenübergreifende Leistungsfähigkeit des visuellen Systems zu demonstrieren, ist der Vergleich der Erkennungsleistung auf der ORL-Gesichtsdatenbank (vgl. auch Abbildung 3.3 auf Seite 58). Die hier enthaltenen Objekte – die Gesichter – haben im Gegensatz zu den COILselect Objekten eine wesentliche geringere Ähnlichkeit zu den COIL20 Objekten. Das beste von Lawrence et al. [23] erreichte Ergebnis bei zwei verwendeten Trainingsansichten ist eine Fehlklassifikationsrate von 17.0%. Das beste visuelle System, das mit Hilfe der indirekten Kodierung optimiert wurde, erreicht eine Fehlklassifikationsrate von 15.3%. Dagegen konnte bei dem evolutionären Entwurfsprozess mit direkter Kodierung der Fehler des visuellen Systems auf 13.8% reduziert werden.

[4]Zu beachten ist, dass die ersten 20 Objekte der COIL100 Datenbank eine andere Schwierigkeit bei der Klassifikation aufweisen können als die 20 Objekte der COIL20 Datenbank.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Ziel der vorliegenden Arbeit war der Aufbau eines Entwurfsprozesses für ein biologisch inspiriertes visuelles Erkennungssystem, das eine Erkennung von realen Objekten anhand weniger Trainingsansichten ermöglicht.

Hierzu wurde ein direkt kodiertes evolutionäres Entwurfsverfahren aufgebaut, das parallelisiert auf einer verteilten Rechnerstruktur genutzt wurde. Dieses Verfahren optimiert wesentliche strukturbestimmende Nichtlinearitäten und entwirft eine Anzahl von höheren Merkmalen des visuellen hierarchischen Systems. Die Merkmale definieren komplexe Kombinationen von Kantenmerkmalen und werden daher auch als Kombinationsmerkmale bezeichnet. Mit Hilfe der Optimierung konnte die Fehlklassifikationsrate des Ausgangssystems [49] von 14.4% auf 6.5% verringert werden. Das entspricht einer Verbesserung um über 50%. Gleichzeitig konnte die benötigte Erkennungszeit halbiert werden. Diese Beschleunigung wurde durch eine Reduktion der benötigten Kombinationsmerkmale von zuvor 50 auf 9 erreicht.

In einem nächsten Schritt wurde ein Evaluierungskonzept definiert, das insbesondere dazu geeignet ist, die Generalisierungsfähigkeit von lernenden Systemen zu beurteilen, die mit Hilfe von Entwurfs- bzw. Optimierungsmethoden aufgebaut wurden. Hierbei werden zwei Arten von Generalisierung unterschieden: die Generalisierung 1. und 2. Ordnung. Erstere bezeichnet das Vermögen des visuellen Systems, von Trainings- auf Testansichten der Objektdatenbank A generalisieren zu können. Die Datenbank A wurde hierbei innerhalb der Strukturoptimierung des visuellen Systems zur Ermittlung seiner Güte bzw. Fitness verwendet. Die Generalisierung 2. Ordnung hingegen drückt die Leistungsfähigkeit des Systems aus, von Trainings- auf Testansichten einer Objektdatenbank B generalisieren zu können, wobei die Datenbank B nicht am Entwurfsprozess des Systems beteiligt war. Die Analyse der optimierten visuellen Systeme ergab, dass neben der bereits erwähnten Verbesserung der Erkennungsleistung auf einer Datenbank A ($\hat{=}$ Generalisierung 1. Ordnung)

um 50% gegenüber dem Ausgangssystem gleichzeitig auch die Generalisierung 2. Ordnung um über 20% verbessert werden konnte.

Durch die Analyse der evolutionär optimierten visuellen Systeme konnten Einsichten in die mögliche Funktionsweise der biologisch motivierten hierarchischen Erkennungsstruktur gewonnen werden. Bei der Untersuchung der Systemnichtlinearitäten wurde die Herausbildung von zwei unterschiedlichen „Strategien" beobachtet. Eine Strategie bildet eine hohe Selektivität des visuellen Inputs mit Hilfe einer hohen lateralen Kompetition heraus, gefolgt von einer geringen Selektivität mittels eines kleinen Schwellwertes. Die zweite Strategie implementiert den genau umgekehrten Weg, um eine gewünschte Selektivität festzulegen.

Zur Analyse der Kombinationsmerkmalsbänke wurde ein spezielles Abstandsmaß vorgeschlagen, das den quantitativen Vergleich unterschiedlicher Merkmalsbänke erlaubt. Es ergibt sich, dass die durch die Evolution gefundenen Merkmalsbänke, die einer ähnlich guten Fitness zugeordnet sind, einen großen Abstand voneinander haben. Das spricht für die robuste Struktur des visuellen Systems, das bei einer sehr großen Anzahl von unterschiedlichen Kombinationsmerkmalen eine gute Erkennungsleistung aufweist.

Zu einer weiteren Erhöhung der Robustheit des visuellen Systems, insbesondere in Bezug auf die Generalisierung 2. Ordnung, wurden zwei Methoden vorgeschlagen und untersucht. Die erste Methode beruht auf einer Regularisierung durch eine Verallgemeinerung des bisher verwendeten Fitnessmaßes (= negative Fehlklassifikationsrate). Hierbei werden kontinuierliche positive Beiträge bei einer korrekten Klassifikation und negative Beiträge bei einer Fehlklassifikation verwendet. Die Höhe des Beitrags richtet sich u.a. nach der Größe einer Sicherheitsmarge, mit der die Entscheidung gefällt wird. Die Untersuchung ergab, dass die Einführung dieses veränderten Fitnessmaßes zwar zu einer kleinen, aber nicht signifikanten Erhöhung der Generalisierung 2. Ordnung führt. Dies kann damit erklärt werden, dass der positive Effekt des Zugewinns an Robustheit nicht groß genug im Verhältnis zu dem negativen Effekt der geringeren Gewichtung der Klassifikationsrate innerhalb der Optimierung ist.

Die zweite vorgeschlagene Methode zur Erhöhung der Generalisierung 2. Ordnung sieht die Implementation einer veränderlichen visuellen Umgebung bzw. Aufgabe vor. Hierzu werden nicht mehr alle Objekte der Datenbank A gleichzeitig benutzt. Vielmehr wird die Güte eines visuellen Systems nur noch an der Klassifikationsleistung auf einer Untermenge aller Objekte gemessen. Die Auswahl der zur Klassifikation verwendeten Objekte wird durch ein Auswahlfenster festgelegt. Dieses wird im Verlauf der evolutionären Optimierung zeitlich verschoben, so dass die Systeme bevorzugt werden, deren Nichtlinearitäten und Merkmale von allgemeinerer Güte und Anwendbarkeit sind. Die durchgeführte Untersuchung zeigte eine signifikante Erhöhung der Generalisierung 2. Ordnung durch die Verwendung des vorgeschlagenen Schemas der veränderlichen Umgebung.

Zur systematischen Untersuchung der Leistungsfähigkeit des biologisch inspirierten visuellen Systems im Zusammenspiel mit dem evolutionären Entwurfskonzept wurde ein sogenanntes generatives Modell vorgestellt. Dieses ermöglicht die Erzeugung von Musterdatenbanken, bestehend aus hierarchisch aufgebauten Mustern. Die Optimierung des visuellen Systems auf einer beispielhaft generierten Musterdatenbank zeigte das Funktionieren der Entwurfsmethode. Außerdem zeigte sich auch hier die große Variabilität optimaler Lösungen. Weiter wurde die Vermutung bestätigt, dass die Generalisierungsfähigkeit des visuellen hierarchischen Systems auf den hierarchischen Daten der eines Multi-Layer-Perceptrons überlegen ist.

Alternativ zu dem Entwurfsverfahren, welches auf einer **direkten** Kodierung innerhalb der evolutionären Optimierung beruht, wurde ein **indirekt** kodiertes Verfahren entwickelt. Mit diesem wird erstmals in dieser Form ein biologisch motiviertes visuelles System anhand einer anspruchsvollen dreidimensionalen Objekterkennungsaufgabe evolutionär optimiert. Die indirekte Kodierung bettet hierbei in Anlehnung an die Ontogenese einen unüberwachten lokalen Lernprozess in den evolutionären Optimierungsvorgang ein. Durch die Integration eines Lernverfahrens in die Genotyp-Phänotyp-Abbildung wird eine enge Kopplung von Evolution und Lernen geschaffen. Diese ermöglicht ein Entwurfsverfahren, das mit einem viel kleineren Genom und damit mit einem viel niedrigdimensionaleren Suchraum auskommt. Für den Einsatz der unüberwachten Lernverfahren wurden unterschiedliche Methoden erprobt. Das beste Resultat erbrachte ein Verfahren, das eine nicht negative überrepräsentierte spärliche Kodierung der Eingangsmuster implementiert.

Abschließend wurde ein Vergleich der beiden unterschiedlichen Entwurfsverfahren, der direkten und der indirekten Kodierung, vorgenommen. Charakteristisch für den Optimierungsverlauf war ein schnelleres Absinken des Klassifikationsfehlers bei der indirekten Kodierung. Verantwortlich hierfür ist die Verwendung des lokalen Lernens bei dem Aufbau jedes einzelnen Individuums (im Rahmen der Genotyp-Phänotyp-Abbildung). Bezogen auf die Struktur der optimierten Kombinationsmerkmalsbänke ergibt sich eine gleichmäßigere Nutzung der Merkmale, die aus der indirekten Kodierung hervorgegangen sind. Das Abstandsmaß bestätigt auch den starken Unterschied der mit Hilfe der direkten und der indirekten Kodierung optimierten Merkmalsbänke.

Um die Leistungsfähigkeit der optimierten visuellen Systeme besser einordnen zu können, wurde ein Vergleich mit gegenwärtigen performanten Erkennungssystemen vorgenommen. Hierbei ergibt sich für das visuelle System eine in vielen Bereichen überlegene Erkennungsrate.

6.2 Ausblick

Aufbauend auf dem entwickelten evolutionären Entwurfsverfahren für das visuelle Erkennungssystem ergeben sich zahlreiche vielversprechende Erweiterungsmöglichkeiten.

Eine Erweiterung des visuellen Systems um zusätzliche parallele Verarbeitungspfade stellt eine natürliche Fortentwicklung dar. So kann das System flexibel beispielsweise um Farb- und weitere Formkanäle unterschiedlicher räumlicher Auflösung verbreitert werden. Die Erweiterung kann entweder zu Beginn des Entwurfsprozesses vorgegeben oder aber mit Hilfe von Wachstumsoperatoren dynamisch in das evolutionäre Konzept eingebunden werden. Auf diese Weise kann durch die Integration von Wachstum die Ausbildung bestimmter Merkmale unter Umständen noch bedarfsgerechter implementiert werden. Das heißt, die neu hinzukommenden Kanäle werden das System um diejenigen Merkmalsunterräume erweitern, die aus der bisherigen Entwicklung des Systems am erfolgversprechendsten erscheinen. Zu beachten ist hierbei, dass nach dem Wachstum großer strukturell neuer Bestandteile zunächst in der Regel eine Fitnessverschlechterung eintritt. Diese führt dann zu dem Aussterben der entsprechenden Individuen, obwohl die Erweiterung nach weiteren eher graduellen Anpassungen durchaus sinnvoll gewesen wäre. Um solch eine Situation zu vermeiden, gibt es zwei Möglichkeiten. Zum einen können nach großen strukturellen Mutationen Unterpopulationen, sogenannte Nischen, angelegt werden, die dann nach einigen Generationen wieder vereinigt werden können. Zum anderen können die strukturellen Veränderungen zunächst neutral ausgelegt werden. Die Systemstruktur muss hierzu so aufgebaut sein, dass auch strukturelle Änderungen so parametrisierbar sind, dass ihre Fitnessauswirkungen anfangs zu Null geregelt werden können.

Neben der Erweiterung der Verarbeitungskapazitäten des visuellen Systems bietet sich mit zunehmender Mächtigkeit auch die Erweiterung der Aufgabenstellung des Systems an. Hierbei soll es nicht um eine rein quantitative Ausweitung, wie die Hinzunahme weiterer Objekte gehen, sondern vielmehr um die qualitative Erweiterung zur Kategorisierung von Objekten. Diese kognitive Leistung geht weit über das reine Wiedererkennen von Objekten hinaus. So sollen unterschiedliche Objekte in eine übergeordnete Kategorie zusammengefasst werden. Die Gesichtspunkte, nach denen Kategorien gebildet werden könnten, sind vielfältig. Die menschliche Kategorienbildung ist ein hoch komplexer und dynamischer Prozess, der stark mit dem „Weltwissen" und dem situativen und funktionalen Kontext verbunden ist. So ordnet der Mensch Objekte z.B. nach Größe, Farbe, Form, Material, Oberflächenbeschaffenheit, möglichen Funktionen und nach vielen anderen Aspekten. Das entsprechende evolutionäre Entwurfsverfahren kann auf unterschiedliche Arten gestaltet werden: Es kann einerseits eine Kategorisierung vorgegeben werden und das System wird dahingehend optimiert, die gegebenen Objekte durch die Ausbil-

dung entsprechender komplexer Merkmale zuzuordnen. Oder das System kann andererseits unüberwacht eigene Kategorisierungen ausbilden, die im Nachhinein interpretiert werden können.

Im letzteren Fall könnte ein Wachstumskriterium für das System darin liegen, dass nach Möglichkeit nicht zu viele Objekte in eine Kategorie eingeordnet werden sollen. Man könnte sich vorstellen, dass die evolutionäre Optimierung dynamisch so organisiert ist, dass im Zeitverlauf immer weitere Objekte dem visuellen System präsentiert werden. Steigt dabei die Anzahl der Objekte innerhalb einer Kategorie über eine bestimmte Schwelle, so wird ein Merkmalswachstum getriggert, das zu einer Unteraufteilung dieser Klasse führt.

Gibt man die gewünschte Aufteilung der Objekte in Kategorien und Unterkategorien vor, so könnte man den evolutionären Entwurfsprozess dazu nutzen, um spezielle – zur Unterscheidung bestimmter Kategorien geeignete – Merkmale und Merkmalspfade aufzubauen. In einem später verwendeten Gesamtsystem könnten diese Informationen dann aktiv genutzt werden. Wenn beispielsweise dem visuellen System als Vorwissen mitgeteilt wird, in welcher momentanen Umgebung es sich befindet, könnte das System entsprechend wenig geeignete Merkmale deaktivieren und wichtige höher bewerten. Die Notwendigkeit von Vorwissen könnte auch durch eine mögliche „top-down"-Information, erzeugt vom visuellen Systems selbst, ersetzt werden. Das bedeutet: nachdem das visuelle System erkannt hat, in welcher visuellen Umgebung es sich befindet, gewichtet es automatisch die für diese Umgebung geeigneten Merkmale stärker.

Die entwickelten evolutionären Entwurfsverfahren sind so allgemein ausgelegt, dass sie sich für vielfältige Erweiterungen sowohl des visuellen Systems als auch der zu optimierenden Aufgabenstellung sehr gut eignen. Damit ist ein Fundament gelegt für den integrierten Entwurf von technischen Objekterkennungssystemen, deren robuste, domänenübergreifende und anpassungsfähige Funktionsweise sich der des menschlichen Sehsystems weiter annähert.

Abkürzungsverzeichnis

AGS	**A**us**g**angs**s**ystem
BP	**B**ack**p**ropagation
ES	**E**volutions**s**trategie
GA	**G**enetischer **A**lgorithmus
GPA	**G**enotyp-**P**hänotyp-**A**bbildung
ICA	**I**ndependent-**C**omponent-**A**nalysis
KM	**K**ombinations**m**erkmal
MLP	**M**ulti-**L**ayer-**P**erceptron
mVS	**M**anuell optimiertes **v**isuelles **S**ystem
NNE	**N**ächster-**N**achbar-**E**rkenner
NNK	**N**ächster-**N**achbar-**K**lassifikator
optVS	Evolutionär **opt**imiertes **v**isuelles **S**ystem
PCA	**P**rinciple-**C**omponent-**A**nalysis
RF	**R**ezeptives **F**eld
SLP	**S**ingle-**L**ayer-**P**erceptron
SVM	**S**upport-**V**ektor-**M**aschine
WTM	**W**inner-**T**ake-**M**ost

Symbolverzeichnis

α	Orientierungswinkel der Gabormerkmale
α_{Test}	Ansichtswinkel der Testansicht
α_{Train}	Ansichtswinkel der Trainingsansicht
$\alpha_{\text{Validierung}}$	Ansichtswinkel der Validierungsansicht
γ_i	Nichtlinearität zur Steuerung der Kompetition innerhalb der Schicht Si
η	Schrittweite zur Adaptation von Strategieparametern
θ_i	Schwellwert der Schicht Si
λ	Größe der Nachkommenpopulation
μ	Größe der Elternpopulation
σ	Standardabweichung
$\sigma_{\text{g}s}$	Globale Mutationsschrittweite
$\boldsymbol{\sigma}_{\text{i}s}$	Mutationsschrittweitenvektor
σ_i	Poolingweite der i-ten Schicht
σ_{dist}	Skalierungsfaktor der Abstandsfunktion dist
σ_{km}	Mutationsschrittweite der Kombinationsmerkmale
σ_{nl}	Mutationsschrittweite der Nichtlinearitäten
σ_{noise}	Standardabweichung des normalverteilten Rauschens
τ, τ_0	Schrittweite zur Adaptation von Strategieparametern
$\bar{\mathbf{c}}_i$	Schichtaktivierung der Schicht Ci
$\mathbf{c}_i^l$	Aktivierung der l-ten Ebene der Schicht Ci
$c_i^l(x,y)$	Aktivierung des Neurons der l-ten Ebene der Schicht Ci an der Position (x,y)
$\bar{\mathbf{c}}_i^{(p)}$	Vektor generiert durch das Ausschneiden des Bildpatches p aus der Schichtaktivierung Ci
$\mathbf{d}$	Zielvektor einer Funktion
$\text{dist}_{nao}(i)$	Abstand der zu klassifizierenden Objektansicht zur nächsten gelernten Ansicht eines anderen Objektes
$\text{dist}_{ngo}(i)$	Abstand der zu klassifizierenden Objektansicht zur nächsten gelernten Ansicht desselben Objektes
$d_i^{KM}(B1,B2)$	Euklidischer Abstand des i-ten Merkmals der Merkmalsbank B1 zu dem nächsten Merkmal der Bank B2

$D_{KM}(B1, B2)$	Distanzmaß der beiden Kombinationsmerkmalsbänke B1 und B2
E	Wert einer zu minimierenden Funktion
F_b	Breite des Auswahlfensters
F_{sw}	Sprungweite des Auswahlfensters
$\mathbf{g}_i$	Generatives Untermodell, das die Schichtaktivierungen $\mathbf{c}_{i-1}$ auf die Aktivierungen $\mathbf{c}_i$ abbildet
$\mathbf{g}_{\text{Gauß},i}(x, y)$	Normalisierte Gaußsche Glockenkurve mit Standardabweichung σ_i
$g_i^{(j)}$	j-tes Untermodell des generativen Untermodells $\mathbf{g}_i$
$\mathbf{G}^H$	Generatives Modell mit H Schichten
G_{sprung}	Verweildauer des Auswahlfensters in Generationen
H	Anzahl der Schichten des generativen Modells
H(.)	Heavysidefunktion
$\mathbf{I}$	Grauwertmatrix des Inputbildes
L	Anzahl der Kombinationsmerkmale
M	Maximalwert
$N(0, \sigma^2)$	Standard Normalverteilung mit Mittelwert 0 und Varianz σ^2
O	Zustandsraum aller möglichen visuellen Inputdaten
O_i	Zustandsraum aller visuellen Inputdaten des Objektes i
p	Index eines Bildpatches
P_i	Anzahl der Ebenen der Schicht i
q	Wichtungsfaktor der kontinuierlichen Fitnessfunktion
$Q_i^{(r)}$	Menge der Aktivierungen der Schicht i, die auf die r-te Aktivierung der Schicht $i+1$ abbilden
$\bar{\mathbf{s}}_i$	Schichtaktivierungsvektor der Schicht Si
$\mathbf{s}_i^l$	Aktivierung der l-ten Ebene der Schicht Si
$s_i^l(x, y)$	Aktivierung des Neurons der l-ten Ebene der Schicht Si an der Position (x, y)
$s_l^{(p)}$	Koeffizient der nnSC-Methode für das l-te Kombinationsmerkmal und den p-ten Patch der C2-Schicht
T	Zustandsraum aller Trainingsansichten; $T \subset O$
T_i	Zustandsraum aller Trainingsansichten des Objektes i
$\mathbf{w}_1^l$	Merkmalsvektor des l-ten Merkmals der 1. Schicht
$\bar{\mathbf{w}}_i^l$	Merkmalsvektor des l-ten Merkmals der i-ten Schicht; für $i = 1$ gilt: $\bar{\mathbf{w}}_1^l = \mathbf{w}_1^l$
$\mathbf{w}_2^{lk}$	Gewichtsvektor des l-ten Merkmals der k-ten Ebene der Schicht S2
$\mathbf{w}_2^{lk}(x, y)$	Rezeptiver-Feld-Vektor des l-ten Merkmals des S2-Neurons an der Stelle (x, y); er enthält die Gewichte zu der k-ten Ebene der vorangehenden Schicht C1
w_{kij}	Netzgewicht der Verbindung des i-ten Neurons der Schicht $k-1$ zum j-ten Neuron der k-ten Schicht
$\mathbf{x}(t)$	Objektparametervektor in Generation t

$\mathbf{y}$	Ausgangsvektor einer Funktion
$\mathbf{z}$, $\mathbf{z}''$	Vektor, dessen Komponenten normalverteilte Zufallszahlen enthalten
z'	Normalverteilte Zufallszahl

Allgemeine Symbole:

N_X	Anzahl der Elemente der Menge X
$\mathbf{z}^\top$	Vektor $\mathbf{z}$ transponiert
$\mathbf{z}(t)$	Vektor $\mathbf{z}$ in der Generation t
z_i	i-te Komponente des Vektors $\mathbf{z}$
$z_i(t)$	i-te Komponente des Vektors $\mathbf{z}$ in der Generation t

Literaturverzeichnis

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, 2000.

[2] H. B. Barlow. The twelfth Bartlett memorial lecture: The role of single neurons in the psychology of perception. *Quarterly Journal of Experimental Psychology*, 37:121–145, 1985.

[3] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[5] G. Brown, X. Yao, J. Wyatt, H. Wersing, and B. Sendhoff. Exploiting ensemble diversity for automatic feature extraction. In L. Wang, J. Rajapakse, K. Fukushima, S.-Y. Lee, and X. Yao, editors, *Proceedings of the 9th International Conference on Neural Information Processing - ICONIP*, volume 4, pages 1786–1790, 2002.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, volume November. John Wily & Sons, Inc., New York, second edition, 2000.

[7] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks – a review. *Pattern Recognition*, 35(10):2279–2301, 2002.

[8] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 39:139–202, 1980.

[9] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[10] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, Nov. 1994.

[11] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, Piscataway, NJ, 1996. IEEE Service Center.

[12] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[13] S. Haykin. *Neural Networks.* Macmillan College Publishing Company, New York, 1994.

[14] J. Hegdé and D. Van Essen. Selectivity for complex shapes in primate visual area V2. *The Journal of Neuroscience*, 20:RC61(1–6), 2000.

[15] G. Hinton and S. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.

[16] V. Honavar and L. Uhr. Brain-structured connectionist networks that perceive and learn. *Connection Science*, 1:139–160, 1989.

[17] P. O. Hoyer and A. Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002.

[18] D. Hubel and T. Wiesel. Receptive fields of single neurons in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1959.

[19] M. Hüsken, J. Gayko, and B. Sendhoff. Optimization for problem classes - neural networks that learn to learn. In X.Yao, editor, *IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks.* IEEE Press, 2000. 98-109.

[20] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[21] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.

[22] E. Körner, M.-O. Gewaltig, U. Körner, A. Richter, and T. Rodemann. A model of computation in neocortical architecture. *Neural Networks*, 12(7-8):989–1005, 1999.

[23] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[24] M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525 – 533, 1993.

[25] S. K. Nayar, S. A. Nene, and H. Murase. Real-time 100 object recognition system. In *Proceedings of ARPA Image Understanding Workshop*, Palm Springs, 1996.

[26] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[27] Z. Pan, T. Sabisch, R. Adams, and H. Bolouri. Staged training of neocognitron by evolutionary algorithms. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1965–1972. IEEE Press, 1999.

[28] S. Quartz and T. Sejnowski. The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences*, 9:537–596, 1997.

[29] I. Rechenberg. *Evolutionsstrategie '94*. Friedrich Frommann Holzboog Verlag, Stuttgart, 1994.

[30] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[31] H. Ritter, T. Martinez, and K. Schulten. *Neuronale Netze*. Addison-Wesley, 2. edition, 1991.

[32] E. T. Rolls and S. M. Stringer. On the design of neural networks in the brain by genetic evolution. *Progress in Neurobiology*, 6(61):557–579, 2000.

[33] D. Roobaert and M. V. Hulle. View-based 3d object recognition with support vector machines. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing, Madison,USA*, pages 77–84, New York, USA, 1999. IEEE.

[34] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner. Evolutionary feature design for object recognition with hierarchical networks. In L. Wang, J. Rajapakse, K. Fukushima, S.-Y. Lee, and X. Yao, editors, *Proceedings of the 9th International Conference on Neural Information Processing - ICONIP*, volume 4, pages 1936–1940, 2002.

[35] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner. Coupling of evolution and learning to optimize a hierarchical object recognition model. In X. Yao et al., editor, *Parallel Problem Solving from Nature – PPSN VIII*, Lecture Notes in Computer Science 3242, pages 662–671. Springer, 2004.

[36] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner. Evolution of hierarchical features for visual object recognition. In H.-M. Groß, K. Debes, and

H.-J. Böhme, editors, *Third Workshop on Self-Organization of Adaptive Behavior (SOAVE 2004) Ilmenau*, pages 104–113, VDI-Verlag Düsseldorf, 2004. Fortschrittsberichte des VDI.

[37] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner. Evolutionary optimization of a hierarchical object recognition model. *IEEE Trans. Systems, Man, Cybernetics, Part B: Cybernetics*, Special Issue on Learning in Computer Vision and Pattern Recognition, 2004. Submitted.

[38] P. Schuster and P. Stadler. Sequence redundancy in biopolymers: A study on RNA and protein structures. In G. Myers, editor, *Viral Regulatory Structures*, volume XXVIII of Santa Fe Institute Studies in the Sciences of Complexity, Reading MA, 1997. Addison-Wesley.

[39] H. Schwefel. *Evolution and Optimum Seeking.* John Wiley & sons, New York, 1995.

[40] H.-P. Schwefel. Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. In *Interdisciplinary System Research*, Basel, 1977. Birkhäuser.

[41] H.-P. Schwefel and G. Rudolph. Contemporary evolution strategies. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Proceedings of the Third European Conference on Artificial Life : Advances in Artificial Life*, volume 929 of *LNAI*, pages 893–907, Berlin, June 1995. Springer Verlag.

[42] B. Sendhoff and M. Kreutz. A model for the dynamic interaction between evolution and learning. *Neural Processing Letters*, 10(3):181–193, 1999.

[43] B. Sendhoff, M. Kreutz, and W. von Seelen. A condition for the genotype–phenotype mapping: Causality. In T. Bäck, editor, *Genetic Algorithms: Proceedings of the 7th International Conferences (ICGA)*, pages 73–80. Morgan Kaufmann, 1997.

[44] D. Shi, D. Chunlei, and Y. Daniel S. Neocognitron's parameter tuning by genetic algorithms. *International Journal of Neural Systems*, 9:497–509, 1999.

[45] M.-Y. Teo, L.-P. Khoo, and S.-K. Sim. Application of genetic algorithms to optimise neocognitron network parameters. *Neural Network World*, 7(3):293–304, 1997.

[46] M.-Y. Teo and S.-K. Sim. Training the neocognitron network using design of experiments. *Artificial Intelligence in Engineering*, 9(2):85–94, 1995.

[47] H. Ulyings, K. Kuypers, M. Diamond, and W. Veltman. Effects of differential environments on plasticity of dendrites of cortical pyramidal neurons in adult rats. *Experimental Neurobiology*, 62:658–677, 1978.

[48] R. L. D. Valois and K. K. D. Valois. *Spatial Vision*. Oxford University Press, Oxford, UK, 1988.

[49] H. Wersing and E. Körner. Unsupervised learning of combination features for hierarchical recognition models. In J. R. Dorronsoro, editor, *International Conference of Artifical Neural Networks ICANN*, pages 1225–1230. Springer, 2002.

[50] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant recognition. *Neural Computation*, 15(7):1559–1588, 2003.

[51] R. O. L. Wong. Retinal waves and visual system development. *Annual Review of Neuroscience*, 22:29–47, 1999.

[52] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.