Neuronale Fuzzy-Systeme

Rudolf Kruse und Detlef Nauck
Institut für Betriebssysteme und Rechnerverbund
Technische Universität Braunschweig
Bültenweg 74/75, 38106 Braunschweig
Tel. (0531) 391-3155, Fax: (0531) 391-5936
E-Mail: R.Kruse@tu-bs.de bzw. D.Nauck@tu-bs.de, WWW: http://www.cs.tu-bs.de/ibr/fuzzy

1 Einleitung

Stellen Sie sich vor, Sie sollten eine automatische Regulierung der Temperatur für Ihr Wohnzimmer auf einem Personal Computer entwickeln. Die Beschreibung Ihrer Problemlösung in natürlicher Sprache würde Ihnen keinerlei Schwierigkeiten bereiten. Zum Beispiel könnten Sie eine Reihe von Vorschriften festlegen, wie „Wenn die Temperatur *ungefähr 19 Grad* beträgt, dann muß das Heizungsventil *ein wenig* geöffnet werden", „bei *etwa 24 Grad* sollte es *fast* geschlossen werden", usw. Wie können jedoch diese bewußt vage formulierten Regeln formal dargestellt werden, damit sie einer maschinellen Verarbeitung zugänglich sind?

Die einfachste Möglichkeit, nicht-präzise Anweisungen in Rechnerprogramme umzusetzen, besteht darin, Methoden der Fuzzy-Logik zu verwenden [Kruse et al., 1994a, Kruse et al., 1995a]; diese haben sich mittlerweile in vielen industriellen Anwendungen bewährt. Bei der Umsetzung der oben beispielhaft genannten *linguistischen Regeln* (*Fuzzy-Regeln*), sind die umgangssprachlichen Ausdrücke, die sogenannten *linguistischen Terme*, wie *ungefähr 19 Grad*, *ein wenig* usw. durch *Fuzzy-Mengen* zu modellieren. Dieser Modellierungsprozeß ist stark kontextabängig und subjektiv. Verhält sich das resultierende *Fuzzy-System* nicht wie gewünscht, muß die Modellierung nachgebessert werden („tuning"). Dafür fehlen bisher allerdings formale Grundlagen, so daß man letztendlich auf eine durch Versuch und Irrtum geprägte Vorgehensweise angewiesen ist.

Besser wäre es, wenn die Modellierung gar nicht selbst vorgenommen werden müsste, oder wenn eine anfängliche Modellierung automatisch optimiert werden könnte. Dazu wird ein Verfahren benötigt, das in der Lage ist, die Fuzzy-Mengen und gegebenenfalls auch die linguistischen Regeln selbst zu erlernen. Da Fuzzy-Regeln in leicht verständlicher Form angegeben werden, kann das Lernergebnis anschließend in einfacher Weise auf seine Plausibilität hin überprüft werden.

Eine Möglichkeit des maschinellen Lernens, bieten die künstlichen Neuronalen Netze [Nauck et al., 1996, Rojas, 1993]. Ihr Nachteil besteht jedoch darin, daß sich ihr Erlerntes einem Bediener in der Regel nicht erschließt und bestehendes Wissen sich schlecht in ein Neuronales Netz integrieren läßt. Ein Regelungssystem für unser Beispielproblem ließe sich leicht anhand von Beispieldaten durch ein Neuronales Netz implementieren. Allerdings verhindert das Black-Box-Verhalten des Netzes eine Plausibilitätsüberprüfung. Weiterhin ist es meist nicht möglich Vorwissen über die Regelung, wie es oben beispielhaft formuliert wurde, a priori als Wissen in das Netz zu integrieren, um so den Lernvorgang abzukürzen.

Es liegt daher nahe, die Vorteile der Lernfähigkeit Neuronaler Netze mit denen der Transparenz von Fuzzy-Systemen zu verbinden und sogenannte *Neuronale Fuzzy-Systeme* [Nauck et al., 1996] zu entwickeln. Neuronale Fuzzy-Systeme sind ein Ansatz, um wissensbasierte Systeme leistungsfähiger und flexibler zu machen, ohne daß ihre Interpretierbarkeit und Transparenz darunter leidet. Sie zählen zu dem Gebiet „Soft Computing“ bzw. „Computational Intelligence“, in dem interdisziplinäre Ansätze der Informationsverarbeitung untersucht werden, die Vagheit und Unsicherheit bewußt tolerieren und ausnutzen, um auf diese Weise zu effizienten, handhabbaren und insbesondere preisgünstigen Problemlösungen zu gelangen.

2 Fuzzy-Systeme

Im Jahr 1965 hat der amerikanische Wissenschaftler Lotfi A. Zadeh die Idee der *Fuzzy-Menge* (fuzzy = vage, unscharf, verschwommen) als eine Möglichkeit zur Modellierung umgangssprachlicher Begriffe wie *groß, klein, ungefähr 19 Grad, wenig geöffnet*, usw. vorgestellt [Zadeh, 1965]. Derartige Begriffe werden als *unscharf* oder *vage* bezeichnet, da sie nicht einen einzelnen exakten (*scharfen*) Wert repräsentieren, sondern eine Vielzahl von Werten, wobei einige dieser Werte dem jeweiligen Begriff mehr oder weniger gut entsprechen. Zadeh zeigte, wie man mit Hilfe von Fuzzy-Mengen umgangssprachlich formulierte Regeln wie zum Beispiel *Wenn die Temperatur etwas hoch ist, dann schließe das Ventil ein wenig* formal darstellen und verarbeiten kann. Das ist ein wesentlicher Fortschritt gegenüber der Verarbeitung von Regeln mit exakten Werten, die solche Zusammenhänge nur ungenügend abbilden können [Kruse et al., 1994a, Kruse et al., 1995a].

Die erfolgreichsten Anwendungen von „Fuzzy-Systemen“ finden sich im Bereich der Steuerungs- und Regelungstechnik, bei der es um die Beeinflussung von Systemen geht mit dem Ziel, Ausgangsgrößen einen gewünschten zeitlichen Verlauf aufzuprägen. Dabei kann es sich zum Beispiel um die Temperatur einer Raumheizung, den Kurs und die Geschwindigkeit eines Schiffs oder die Konzentration eines Stoffes in einem Bioreaktor handeln. Um zu erreichen, daß die Ausgangsgröße einen Sollwert annimmt, werden Stellgrößen verwendet, die direkt manipulierbar sind, wie etwa die Ventilstellung einer Heizung, die Leistung eines Schiffsmotors oder der Nährstoffzufluß in einem Bioreaktor. Der Wert einer Stellgröße muß ständig so angepaßt werden, daß die Ausgangsgröße möglichst wenig von dem gewünschten Sollwert abweicht. Die Bestimmung des Wertes einer Stellgröße erfolgt auf der Basis aktuell gemessener oder errechneter Eingangsgrößen; oft sind das die Differenz zwischen Soll- und Ausgangswert und die Änderung dieser Differenz.

Als Grundlage für den Entwurf eines Reglers mit Methoden der klassischen Regelungstechnik dient ein mathematisches Modell, meist in Form von Differentialgleichungen, die das dynamische Verhalten des Systems in Abhängigkeit von den Eingangs- und den Stellgrößen möglichst exakt nachbilden. Im Rahmen der Fuzzy-Regelung verzichtet man auf das Erstellen eines mathematischen Modells. Statt dessen versucht man, eine Regelstrategie direkt zu beschreiben, die zum Beispiel durch die Befragung oder Beobachtung eines kompetenten Anlagenfahrers gewonnen wird, der das technische System bedienen kann. Hinter diesem Ansatz steht die Einsicht, daß Menschen meistens in der Lage sind, sowohl einfache als auch komplexe Systeme recht gut zu beherrschen, ohne daß sie dazu ein mathematisches Modell aufstellen müßten. So kann ein Fahrradfahrer eine Kurve fahren, ohne die geringste

Vorstellung zu haben, welche Differentialgleichungen das Fahrverhalten beschreiben.

Ein Fuzzy-Regler benötigt eine Wissensbasis, in der die Regelstrategie in Form geeigneter umgangssprachlicher Regeln festgelegt wird, zum Beispiel *Wenn die Differenz zwischen Soll- und Ausgangswert und die Änderung dieser Differenz ungefähr null sind, dann sollte die Stellgröße in etwa konstant gehalten werden.*

Die in den Fuzzy-Regeln verwendeten linguistischen Terme legen einen unscharfen Bereich oder Zahlenwert fest, der innerhalb des Fuzzy-Reglers durch eine Fuzzy-Menge repräsentiert wird. Im Unterschied zu einer gewöhnlichen Menge, bei der ein Objekt entweder vollständig zur Menge gehört oder kein Element der Menge ist, wird bei Fuzzy-Mengen jedem Element ein *Zugehörigkeitsgrad* zwischen 0 und 1 zugeordnet, der jedoch keinesfalls als Wahrscheinlichkeit, sondern nur im Sinne einer graduellen Erfülltheit interpretiert werden darf.

Wenn beispielsweise die Zahlen $1, \ldots, 100$ die Grundmenge der Objekte darstellt und man sich für die Zahlen (Objekte) interessiert, die die Eigenschaften, eine gerade Zahl zu sein, erfüllen, so wird dies durch die Menge

$$G = \{2, 4, \ldots, 98, 100\}$$

oder äquivalent durch das Prädikat

$$P(x) \Leftrightarrow 2 \text{ teilt } x$$

repräsentiert. Beide Arten der Darstellung sind gleichwertig, denn es gilt offenbar

$$x \in G \Leftrightarrow P(x)$$

für alle $x \in \{1, \ldots, 100\}$. Eine weitere Möglichkeit, die geraden Zahlen zwischen 1 und 100 zu beschreiben, besteht in der Angabe der *charakteristischen Funktion* der Menge G:

$$\mathbb{1}_G \quad : \quad \{1, \ldots, 100\} \to \{0, 1\},$$

$$\mathbb{1}_G(n) \quad \stackrel{d}{=} \quad \begin{cases} 1 & \text{falls } n \text{ gerade,} \\ 0 & \text{sonst.} \end{cases}$$

Der Wert $\mathbb{1}_G(n) = 1$ besagt, daß n zur Menge G gehört, während $\mathbb{1}_G(n) = 0$ dafür steht, daß n nicht in G enthalten ist.

Will man jedoch im Kontext erwachsener deutscher Männer für alle Größenangaben aus $\mathbb{R}$ das vage Prädikat *groß* beschreiben, so ist ein scharfe Menge dazu ungeeignet. Ein Beispiel ist in Abb. 1 zu sehen. Werden z.B. Männer mit einer Körpergröße von etwa 170 cm als *groß* angesehen, so ist die Modellierung mit einer scharfen Menge nicht intuitiv: Personen mit einer Größe von 169 cm würden noch als klein, solche mit 170 cm Größe bereits als groß gelten. Eine intuitivere Modellierung gelingt mit einer unscharfen Menge, die durch eine „verallgemeinerte" charakteristische Funktion repräsentiert wird und auch in Abb. 1 zu sehen ist. Diese *Zugehörigkeitsfunktion* ist rein subjektiv gewählt und kann mit wechselndem Kontext (z.B. erwachsene deutsche Frauen) bzw. wechselndem „Experten" (z.B. Arzt, Bekleidungshersteller etc.) anders ausfallen.

Jedem Wert x der Körpergröße wird ein Zugehörigkeitsgrad zugeordnet, z.B. der Größe 1.50 m der Wert 0.7. Das soll heißen, daß die Größe 1.50 m auf einer Skala von 0 bis 1 mit dem Zugehörigkeitsgrad 0.7 das Prädikat *groß* erfüllt. Je näher der Zugehörigkeitsgrad $\mu_{\text{groß}}(x)$ bei 1 liegt, desto mehr genügt x dem Prädikat *groß*.

Definition 1 *Eine Fuzzy-Menge μ von X ist eine Funktion von der Grundmenge X in das Einheitsintervall, d.h. $\mu : X \to [0,1]$. Mit $\mathcal{F}(X) = [0,1]^X$ bezeichnen wir die Menge aller Fuzzy-Mengen von X.*

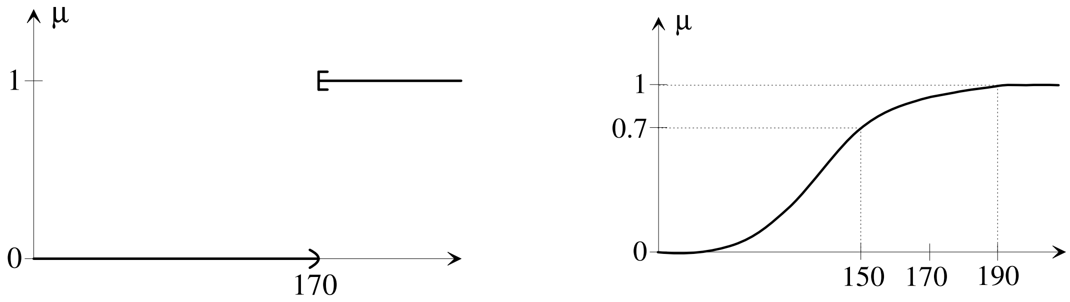

Abbildung 1: Die (scharfe) Menge aller reellen Zahlen ≥ 170 und eine mögliche charakteristische Funktion zur Modellierung des vagen Prädikats *groß*

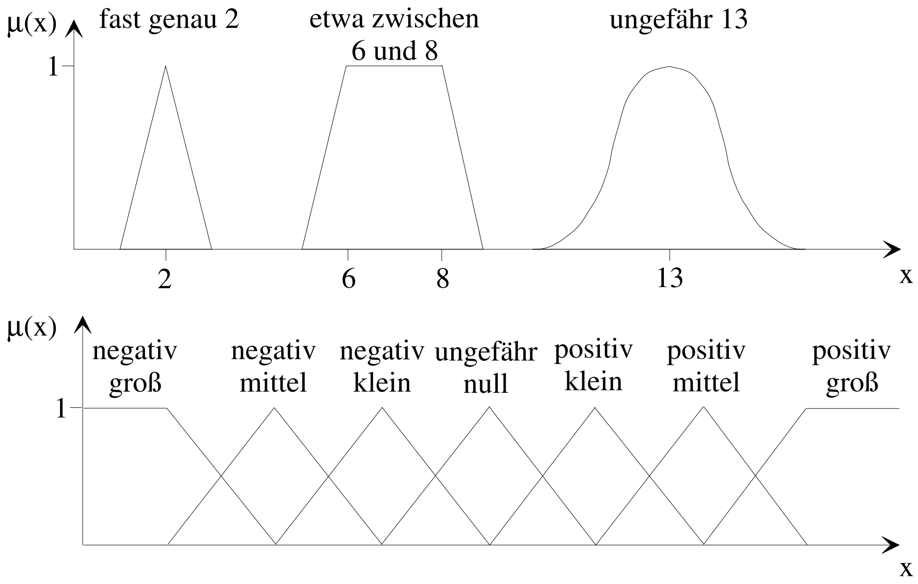

Abbildung 2: Typische Formenen für Zugehörigkeitsfunktionen und eine Fuzzy-Partition aus Dreiecksfunktionen

Zugehörigkeitsgrade erlauben eine realistischere Modellierung vager Konzepte wie *ungefähr* 13 als eine gewöhnliche Menge, bei der eine feste Toleranzschranke angegeben werden müßte, ab der ein Wert nicht mehr als *ungefähr* 13 akzeptiert wird. Eine Fuzzy-Menge ermöglicht einen weichen Übergang von den Werten, die mit Sicherheit als *ungefähr* 13 angesehen werden dürfen, zu denen, die auszuschließen sind. Der Zugehörigkeitsgrad zur *ungefähr* 13 repräsentierenden Fuzzy-Menge wird mit zunehmendem Abstand von 13 kleiner. Üblicherweise beschränkt man sich auf einfache Formen von Fuzzy-Mengen bzw. Zugehörigkeitsfunktionen wie Dreiecks-, Trapez- oder Gaußfunktionen, die sich mit wenigen Parametern festlegen lassen (Abb. 2).

Die in der Wissensbasis eines Fuzzy-Reglers gespeicherten Regeln enthalten für jede Variable — die Eingangsgrößen und im allgemeinen nur eine Stellgröße — verschiedene linguistische

Werte. Um zu garantieren, daß für jeden Wert, den eine Variable annehmen kann, auch mindestens eine passende Regel in der Wissensbasis vorhanden ist, sollten die den linguistischen Werten zugeordneten Fuzzy-Mengen den gesamten Wertebereich der jeweiligen Variablen vollständig überdecken. Man spricht daher auch von einer *Fuzzy-Partition* (Abb. 2).

Das Ziel beim Entwurf eines Fuzzy-Reglers ist die Festlegung eines Kennfeldes, das das Ein-/Ausgabeverhalten des Reglers beschreibt, indem zu jeder möglichen Belegung der Eingangsgrößen ein geeigneter Stellwert angegeben wird. Jede einzelne Regel der Wissensbasis beschreibt den Zusammenhang zwischen Eingangsgrößen und Stellgröße in einem unscharfen Bereich. Auf diese Weise erhält man eine *Fuzzy-Relation*, die die Abhängigkeit der Stellgröße von den Eingangsgrößen vage beschreibt. Eine Fuzzy-Relation ist eine Fuzzy-Menge über dem Produktraum der Stell- und der Eingangsgrößen.

Liegen die aktuellen Werte für die Eingangsgrößen vor, wird für jede Regel mit Hilfe einer t-Norm ihr Erfüllungsgrad ermittelt.

Definition 2 *Eine Funktion* $\top : [0,1]^2 \to [0,1]$ *heißt* **t-Norm**, *wenn sie die Bedingungen*

(i)	$\top(a,1) = a$	*(Einselement)*
(ii)	$a \leq b \Longrightarrow \top(a,c) \leq \top(b,c)$	*(Monotonie)*
(iii)	$\top(a,b) = \top(b,a)$	*(Kommutativität)*
(iv)	$\top(a,\top(b,c)) = \top(\top(a,b),c)$	*(Assoziativität)*

erfüllt.

Die einzelnen Ausgaben der Regeln werden mittels einer t-Conorm zusammengefaßt. Eine t-Conorm ist die zur t-Norm duale Funktion. Sie hat die 0 als Einselement und erfüllt auch die Bedingungen ii – iv der Definition 2. Ein häufig verwendetes Paar aus t-Norm und t-Conorm sind die Minimums- und Maximumsbildung.

Beim sogenannten *Mamdani-Regler* bildet man das Minimum der Zugehörigkeitsgrade der aktuellen Werte zu den in der entsprechenden Regel auftretenden linguistischen Termen bzw. Fuzzy-Mengen. Der Zugehörigkeitsgrad wird dann als Wichtung des unscharfen Ausgabewertes dieser Regel verwendet (Abb. 3). Bei Fuzzy-Reglern vom *Sugeno-Typ* gibt man anstelle der linguistischen Werte für die Stellgröße scharfe Werte vor, so daß auf Fuzzy-Mengen für die Stellgröße verzichtet werden kann. Ein Fuzzy-Regler, der auf der Basis unscharfer Regeln arbeitet, liefert zunächst auch nur ein unscharfes Ergebnis in Form einer Fuzzy-Menge, die noch „defuzzifiziert“ werden muß, um einen eindeutigen Wert für die Stellgröße zu erhalten. Eine häufig angewandte heuristische Defuzzifikations-Strategie ist die Wahl des Stellwertes unter dem Schwerpunkt der durch die Fuzzy-Menge bzw. Zugehörigkeitsfunktion begrenzten Fläche. Dadurch haben die Werte mit geringerem Zugehörigkeitsgrad einen kleineren Einfluß auf den defuzzifizierten Wert, werden aber trotzdem berücksichtigt.

So plausibel das den Fuzzy-Reglern zugrundeliegende Prinzip auch klingen mag, der Teufel bei ihrer Realisierung steckt im Detail. Der erste Entwurf der Regelbasis und der entsprechenden Fuzzy-Partitionen führt in den Anwendungen häufig sehr schnell zu einer Regelung, die das System in groben Zügen unter Kontrolle bringt. Wird eine möglichst genaue und optimale Regelung benötigt, muß aufwendige Feinarbeit — meist mit der Methode „Versuch und Irrtum“ — zur Anpassung der Fuzzy-Mengen, der Wissensbasis und eventuell der eingesetzten mathematischer Operationen (Auswahl der t-Norm, Defuzzifikationsstrategie) geleistet werden. Dies liegt vor allem daran, daß es noch keine formalen Methoden zur

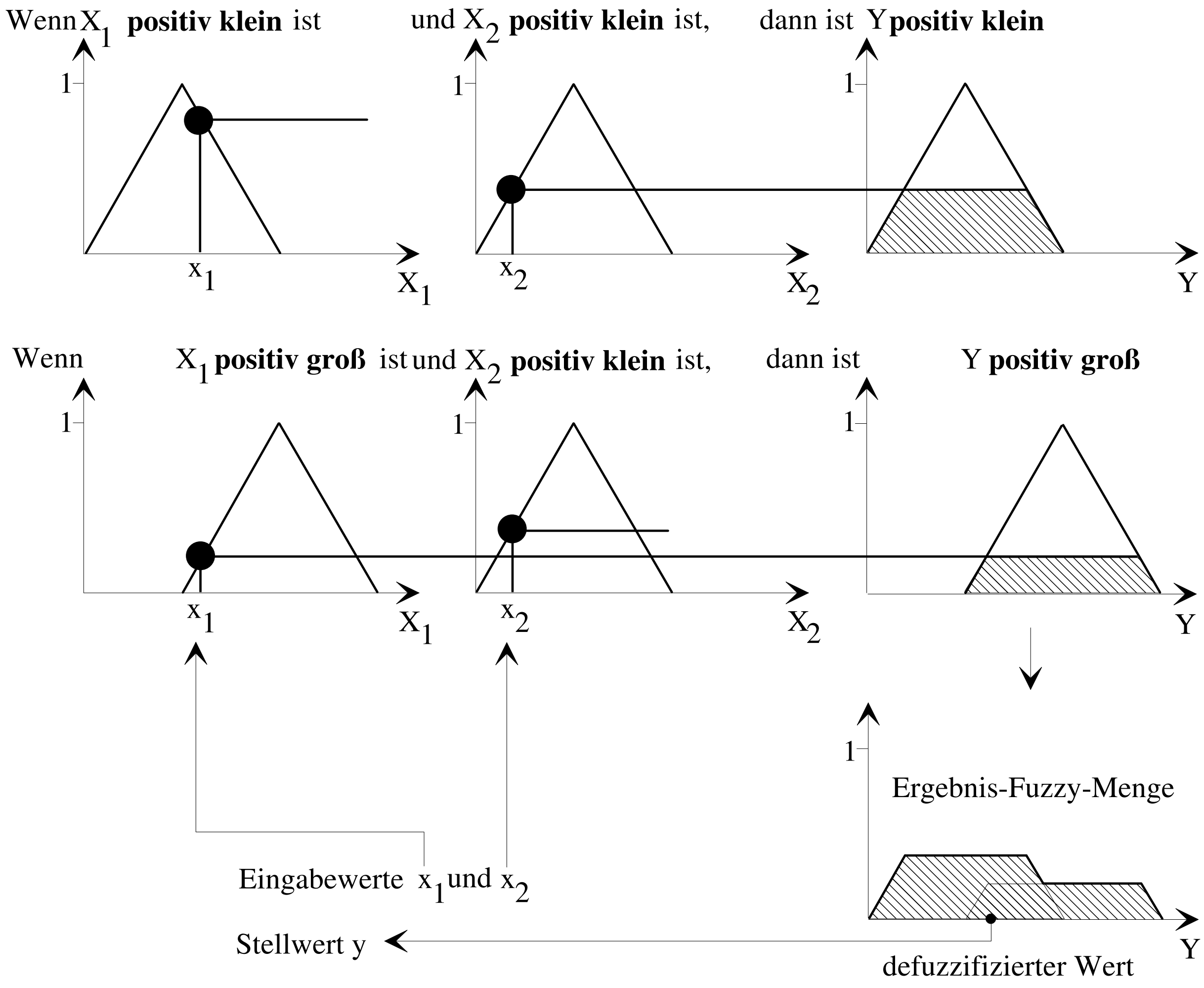

Abbildung 3: Auswertungsschema für einen vereinfachten Fuzzy-Regler mit zwei Eingangs- und einer Ausgangsgrßße und nur zwei Fuzzy-Regeln in der Wissensbasis

Optimierung von Fuzzy-Reglern gibt, und ein Entwickler somit das notwendige „Fingerspitzengefühl" besitzen muß um erfolgreich zu sein. Kopplungen mit Neuronalen Netzen, die aus Beispielen lernen können, bieten hier einen aussichtsreichen Ansatz.

Es soll noch darauf hingewiesen werden, das die hier beschriebenen Fuzzy-Regler strenggenommen keine Methoden der mathematischen mehrwertigen Logik nutzen, auch wenn sie häufig als „Fuzzy Logic Controller" bezeichnet werden. Es haben sich zwei Interpretationen des Begriffs „Fuzzy-Logik" eingebürgert. Man spricht von *Fuzzy-Logik im engeren Sinne*, wenn man sich auf eine Verallgemeinerung der logischen Wahrheitswerte auf das Einheitsintervall bezieht. Die logischen Operationen werden dann durch geeignete t-Normen und t-Conormen verallgemeinert.

Von *Fuzzy-Logik im weiteren Sinn* spricht man im Zusammenhang mit Anwendungen wie z.B. Fuzzy-Reglern. Die eingesetzen Fuzzy-Regeln dürfen hierbei nicht als Regeln im logischen Sinn betrachtet werden. Sie stellen vielmehr eine unscharfe Beschreibung von Punkten in dem Raum der Eingabe- und Ausgabegrößen dar. Anhand dieser unscharfen Stützstellen wird eine unbekannte Funktion partiell beschrieben. Die von einem Fuzzy-Regler vorgenommenen Auswertung bestimmt diese unbekannte Funktion durch ein Interpolationsverfahren in einer unscharfen Umgebung.

Fuzzy-Systeme nach der Art der Fuzzy-Regler approximieren also eine Funktion anhand von Beispielen. Damit ergibt sich eine gewisse Verwandtschaft zu Neuronalen Netzen, die, auf eine andere Weise, ebenfalls Funktionen auf der Grundlage von Beispieldaten approximieren. Fuzzy-Systeme lassen sich daher nicht nur für Regelungsaufgaben einsetzen, sondern auch zu Zwecken der Datenanalyse, wie z.B. Klassifikation und Clustering.

3 Kopplungen von Neuronalen Netzen und Fuzzy-Systemen

Die Idee liegt nahe, Neuronale Netze zur Bestimmung der Parameter eines Fuzzy-Systems heranzuziehen und somit durch eine Kombination dieser beiden Ansätze ein lernfähiges *Neuronales Fuzzy-System* (auch: Neuro-Fuzzy-System) zu schaffen. Auf diese Weise erhält man ein transparentes System, dessen Struktur durch Fuzzy-Regeln geprägt ist und das außerdem adaptiv ist, das heißt, seine Parameter selbständig an seine Aufgabe anpassen kann. Bei diesen Parametern handelt es sich um die Anzahl und Art der Fuzzy-Regeln sowie um die Fuzzy-Mengen, die gewöhnlich durch parametrisierte Dreiecks-, Trapez- oder Gaußfunktionen modelliert werden.

Mittlerweile sind eine ganze Reihe von Neuro-Fuzzy-Systemen entwickelt worden. Es lassen sich sogenannte kooperative und hybride Ansätze unterscheiden. Bei den kooperativen Ansätzen werden Neuronale Netze zur Bestimmung eines Teils der Fuzzy-System-Parameter herangezogen. Nach deren Festlegung wird ein herkömmliches Fuzzy-System erstellt, das ohne das Neuronale Netz arbeitet. Hybride Ansätze „verschmelzen“ sozusagen das Fuzzy-System mit dem Neuronalen Netz. Auf diese Weise entsteht eine Architektur, die sowohl als spezielles Neuronales Netz als auch als gewöhnliches Fuzzy-System interpretiert werden kann. Eine Trennung der beiden Einzelsysteme ist nicht mehr möglich.

Mit dem Begriff „Neuro-Fuzzy“ werden weiterhin auch solche Systeme belegt, die ein Fuzzy-System und ein Neuronales Netz getrennt voneinander zur Bewältigung von Teilen desselben Problems einsetzen. So kann sich ein Neuronales Netz um den Teil kümmern, über den kein Regelwissen vorhanden ist, während das Fuzzy-System das Wissen über den anderen Teil der Problemlösung ausnutzen kann. Nicht immer enthält ein Neuro-Fuzzy-System ein Neuronales Netz in expliziter Form. Vielfach werden lediglich Lernverfahren Neuronaler Netze in ein Fuzzy-System integriert. Einen umfassenden Überblick über zahlreiche Neuro-Fuzzy-Ansätze gibt [Nauck et al., 1996]. Ein einführender Beitrag findet sich in [Kruse et al., 1995b].

Problematisch bei der Erstellung eines Neuro-Fuzzy-Systems kann die Semantik des Modells nach Abschluß des Trainingsvorgangs sein. Ein Ziel des Neuro-Fuzzy-Ansatzes besteht darin, das Modell jederzeit in Form eines herkömmlichen Fuzzy-Systems interpretieren zu können, denn nur so ist die Transparenz des Modells gewährleistet. Werden jedoch Lernalgorithmus und Modellarchitektur nicht sorgfältig aufeinander abgestimmt, so kann es passieren, daß das Lernergebnis semantisch nicht mehr zu deuten ist: Man erhält wie bei Neuronalen Netzen eine Black Box.

Ein wichtiger Aspekt ist daher, daß für einen linguistischen Term wie *niedrige Temperatur* im ganzen System dieselbe Interpretation gilt, das heißt, daß er überall durch die gleiche Fuzzy-

Menge repräsentiert wird. Läßt man zu, daß sich während des Trainings die Fuzzy-Mengen in allen Regeln unabhängig voneinander verändern, so läuft man Gefahr, daß hinterher die eine Regel unter *niedriger Temperatur* etwas anderes versteht als eine andere Regel. Dieses Problem tritt besonders bei der Gewichtung von Regelausgaben auf, die einer unbeschränkten Änderung von Fuzzy-Mengen gleichkommt. Weiterhin möchte man verhindern, daß eine Fuzzy-Menge, die beispielsweise den linguistischen Term *hohe Temperatur* repräsentiert, in einen Bereich verschoben wird, der eigentlich als *niedrige Temperatur* interpretiert wird.

3.1 Kooperative Neuro-Fuzzy-Systeme

Kooperative Neuro-Fuzzy-Systeme können entweder Fuzzy-Regeln oder Fuzzy-Mengen erlernen. Wenn Fuzzy-Regeln gelernt werden sollen, so müssen Fuzzy-Mengen vorgegeben werden, und das System versucht, unter Berücksichtigung dieser Mengen optimale Regeln zu finden. Eine Optimierung von Fuzzy-Mengen setzt dagegen eine feste Fuzzy-Regelbasis voraus. Es lassen sich jedoch unterschiedliche Ansätze nacheinander anwenden. Man kann zum Beispiel Fuzzy-Mengen grob vorgeben und Fuzzy-Regeln erlernen. Anschließend können diese Fuzzy-Mengen mit einem anderen Ansatz weiter optimiert werden.

Die Optimierung von Fuzzy-Mengen läßt sich relativ einfach realisieren. Der Backpropagation-Algorithmus für Neuronale Netze wird auf ein Fuzzy-System mit vollständig gegebener Regelbasis übertragen. Eine schlechte Leistung des Fuzzy-Systems erklärt man durch eine nicht-optimale Modellierung der in den Regeln verwendeten Fuzzy-Mengen. Das Lernverfahren muß daher versuchen, diese Fuzzy-Mengen zu optimieren. Ein derartiges kooperatives Neuro-Fuzzy-Modell enthält also nicht wirklich ein Neuronales Netz, sondern verwendet nur einen neuronalen Lernalgorithmus. Anstelle von Gewichten verändert das Lernverfahren Parameter der Fuzzy-Mengen. Da Backpropagation zu den Gradientenverfahren gehört, setzt es die Differenzierbarkeit der beteiligten Funktionen voraus. Das bedeutet, daß die zur Realisierung der Fuzzy-Mengen und zur Regelauswertung verwendeten Funktionen differenzierbar sein müssen. Man verwendet daher zur Modellierung der Fuzzy-Mengen meist Gaußfunktionen und anstelle der Minimumbildung bei der Regelauswertung die Produktbildung. Für den Lernvorgang wird eine feste Lernaufgabe benötigt, d.h. für gegebene Eingangswerte müssen die dazu passenden Stellwerte bekannt sein. Die mathematischen Probleme, die sich für das Lernen durch eine Maximumsbildung mit anschließender Defuzzifizierung bei der Regelzusammenfassung ergeben würden, werden dadurch ausgeschaltet, daß gewöhnlich nur Fuzzy-Regler nach dem Sugeno-Typ auf diese Weise optimiert werden [Ichihashi, 1991, Nomura et al., 1992].

Fuzzy-Regeln lassen sich zum Beispiel mit Hilfe einer selbstorganisierenden Karte erlernen. Das gewählte Neuronale Netz besteht aus einer Schicht von Eingabeneuronen und einer zweidimensionalen Anordnung von Ausgabeneuronen in der Wettbewerbsschicht — der sogenannten „Karte“. Die Aktivierungen in der Karte zeigen nach dem Lernvorgang an, ob Eingabemuster einander ähnlich sind. In diesem Fall würden sie bei Eingabe in das Netzwerk ein Aktivierungsmaximum an gleicher oder benachbarter Stelle auslösen. Ähnliche Eingaben gehören somit zu einem Cluster, der später als Regel gedeutet wird. Die Aufgabe besteht darin, das Lernergebnis so zu interpretieren, daß diejenigen Cluster gefunden werden, die die Trainingsdaten am besten beschreiben. Wie im Verfahren zum Erlernen von Fuzzy-Mengen, wird auch hier eine feste Lernaufgabe benötigt. Der Unterschied besteht jedoch darin, daß zunächst alle Variablen des Problems Eingaben für das Neuronale Netz darstellen. Eine Aufteilung in Eingangs- und Stellgrößen wird erst nach dem Lernvorgang vorgenommen [Pedrycz und Card, 1992].

Kooperative Ansätze lassen sich außerdem in *offline* und *online* lernende Modelle einteilen. Ein Modell, das offline lernt, benötigt eine feste Lernaufgabe. Für unser eingangs geschildertes Heizproblem wäre das eine Menge von Paaren aus Wohnzimmertemperatur und daraufhin zu veranlassender Ventilöffnung. Das Modell kann nun versuchen, die Werte für die Stellgrößen auf der Grundlage der Eingangsgrößen nachzubilden (überwachtes Lernen), oder es gruppiert zusammengehörige Beispiele in Teilmengen (Cluster), wobei diese Teilmengen vorher nicht bekannt sind (nicht-überwachtes Lernen). Das resultierende Fuzzy-System gelangt erst zum Einsatz, wenn der Lernvorgang abgeschlossen ist.

Ein online lernendes Modell führt dagegen ein „training on the job" durch. Für ein Fuzzy-System wird eine Regelbasis vorgegeben, die zunächst nur grob entworfene Fuzzy-Mengen verwendet. Diese werden in Abhängigkeit der aktuellen Leistung verändert. Das System wird *belohnt,* wenn es sich gut verhält, und *bestraft,* wenn seine Leistung zu wünschen übrig läßt (verstärkendes Lernen). Ein Online-Training ist nur dann möglich, wenn entweder eine Simulation des zu lösenden Problems verfügbar ist (zum Beispiel ein Modell, das aus der Ventilstellung die Wohnzimmertemperatur ableiten kann) oder wenn ein Versagen des Neuro-Fuzzy-Systems keinen Schaden bei der Bearbeitung der realen Aufgabe verursachen kann (beispielsweise könnte Ihr Wellensittich bei zu niedriger Temperatur erfrieren, weil das System noch nicht ausreichend gelernt hat). Ein online lernendes Modell kann ständig weiterlernen und sich so Änderungen der Aufgabe jederzeit anpassen. In diesem Fall werden die zur Optimierung herangezogenen Neuronalen Netze (oder neuronalen Lernverfahren) nicht aus dem System entfernt, wie es sonst bei kooperativen Modellen nach Abschluß des Trainings üblich ist.

Ein sehr einfacher Neuro-Fuzzy-Ansatz, der sehr häufig in kommerziellen Fuzzy-Entwicklungsumgebungen implementiert wird, besteht in der Gewichtung der Fuzzy-Regeln mit einem reellen Wert. Ein Gewicht wird mit der Ausgabe-Fuzzy-Menge einer Regel multipliziert, wodurch die Fuzzy-Menge durch Stauchung oder Streckung modifiziert wird. Die Gewichte können mit Hilfe eines einfachen neuronalen Lernverfahrens (Delta-Regel) adaptiert werden, sofern eine feste Lernaufgabe vorliegt.

Der Nachteil dieses Verfahrens liegt darin, daß die Gewichtung der Regeln zu nicht-normalen Fuzzy-Mengen in der Konklusion führen kann. Liegen die Gewichte in $[0, 1]$ so ergeben sich lediglich Fuzzy-Mengen, die eventuell nirgendwo mehr den Zugehörigkeitsgrad 1 annehmen. Sind die Gewichte allerdings größer 1 oder kleiner 0, so ergeben sich Funktionen, die nicht mehr als Zugehörigkeitsfunktionen interpretiert werden können. Die voneinander unabhängigen Änderungen der Gewichte während eines Lernverfahrens resultieren außerdem häufig darin, daß gleiche linguistische Terme in verschiedenen Regeln unterschiedlich realisert sind.

Auch ist die Interpretation der Gewichte meist unklar. Häufig werden sie als „Regelplausibilitäten" gedeutet, ohne daß jedoch die Semantik dieser Interpretation untersucht wird. Regelgewichte lassen sich durch geeignete Änderungen in den Fuzzy-Mengen der Regelprämissen vermeiden.

3.2 Hybride Neuro-Fuzzy-Systeme

Um mit einem einzigen Neuronalen Fuzzy-System sowohl Fuzzy-Mengen als auch Fuzzy-Regeln zu erlernen, benötigt man ein modernes hybrides Modell. Frühe Ansätze begnügten sich damit, ein System aus spezialisierten Neuronalen Netzen zu bilden, deren Parameter mit denen eines Fuzzy-Systems initialisiert wurden. Ihr Nachteil bestand darin, daß sie nur Fuzzy-Mengen erlernen, und daß die Modelle nach dem Lernvorgang meist nicht mehr als Fuzzy-Systeme interpretiert werden konnten. Modernere Forschungsansätze beruhen dagegen auf dem formalen Modell eines *Fuzzy-Perceptrons*.

Ein Fuzzy-Perceptron ist von einem dreischichtigen Perceptron abgeleitet. Dabei werden die Gewichte auf den Verbindungen durch Fuzzy-Mengen ersetzt (Fuzzy-Gewichte) und die Aktivierungs- und Propagierungsfunktionen so modifiziert, daß sie mit Fuzzy-Mengen operieren können. Die Eingabeeinheiten repräsentieren die Eingangsgrößen, die inneren Einheiten stehen für die Fuzzy-Regeln, und die Ausgabeeinheiten stellen die Stellgrößen dar. Die Propagierungsfunktion zwischen Eingabeschicht und innerer Schicht bestimmt die Zugehörigkeitsgrade der Eingaben zu den Fuzzy-Gewichten der entsprechenden Verbindungen. Durch eine Minimumbildung bestimmt die Aktivierungsfunktion einer inneren Einheit daraus den Erfüllungsgrad einer Regel. Die Propagierungsfunktion zwischen innerer Schicht und Ausgabeschicht verknüpft daraufhin die Erfüllungsgrade mit den Fuzzy-Gewichten der zur Ausgabe führenden Verbindungen. Die Aktivierungsfunktion einer Ausgabeeinheit faßt die ankommenden modifizierten Fuzzy-Mengen zu einer zusammen und führt abschließend eine Defuzzifizierung durch. Damit ein solches Modell immer eindeutig als Fuzzy-System interpretiert werden kann, müssen die oben erwähnten Einschränkungen für den Lernvorgang getroffen werden. *Gekoppelte Verbindungen* stellen sicher, daß gleiche Fuzzy-Mengen sich nicht unterschiedlich entwickeln können.

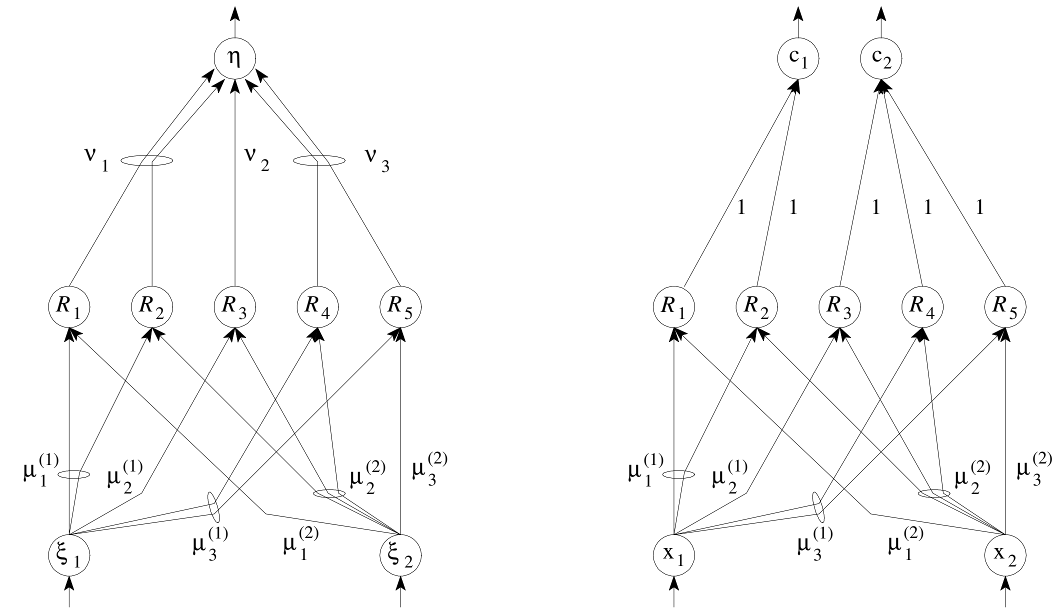

Abbildung 4: Zwei von einem Fuzzy-Perceptron abgeleitete Neuro-Fuzzy-Systeme: Links ein NEFCON-System für regelungstechnische Aufgaben mit Fuzzy-Gewichten in beiden Verbindungsebenen, rechts ein NEFCLASS-System zur Klassifikation von Daten mit Fuzzy-Gewichten nur in der ersten Verbindungsebene. Die Ringe um die Verbindungen stellen gekoppelte Gewichte dar. Diese sind zu jeder Zeit identisch.

Ein derartiges hybrides Neuro-Fuzzy-Modell läßt sich neben der Interpretation als Fuzzy-System auch als Neuronales Netz mit Fuzzy-Gewichten und spezialisierten Aktivierungsund Propagierungsfunktionen ansehen. Dabei gibt man gegenüber herkömmlichen Neuronalen Netzen die Homogenität des Modells (nur gleichartige Neuronen und Funktionen) für die Interpretierbarkeit auf. Die Darstellung und Sicht als Neuronales Netz erlauben eine parallele Implementierung des Systems und gegebenenfalls den Einsatz flexibler „neuronaler Hardware“.

Der Lernalgorithmus zur Anpassung der Fuzzy-Mengen beruht meist auf Variationen des Backpropagation-Verfahrens. Das von uns entwickelte NEFCON-Modell (NEFCON = Neuro Fuzzy Control, s. Abb. 4) nutzt hier eine interessante Variante: Es lernt online durch verstärkendes Lernen und verwendet dazu einen „Fuzzy-Fehler“. Der Benutzer gibt eine Reihe von Regeln an, die beschreiben, wann die Leistung des Systems *gut* und wann sie *schlecht* ist. Ein Beispiel: *Wenn die Temperatur angenehm ist, dann ist die Leistung hoch (der Fehler niedrig)*. Diese Regeln werden wie die eines Fuzzy-Reglers ausgewertet und zur Anpassung der Systemparameter (der Fuzzy-Mengen) herangezogen [Nauck et al., 1996, Nauck und Kruse, 1994, Nauck, 1994].

Für das Erlernen von Fuzzy-Regeln mit einem Neuro-Fuzzy-System gibt es zwei Möglichkeiten: Man beginnt ohne Regeln und nimmt sukzessive Regeln auf, bis die Leistung zufriedenstellend ist, oder man beginnt mit einer Regelbasis aus allen möglichen erzeugbaren Regeln und streicht sukzessive nicht benötigte wieder heraus. Der erste Weg ist weniger aufwendig, kann aber bei online operierendem verstärkenden Lernen gewöhnlich nicht angewandt werden. Eine optimale Ausgabe, die die Konklusion einer neu zu bildenden Regel darstellen würde, ist nicht bekannt, das heißt, die Konklusionen könnten nur geraten werden. Diese Variante bietet sich daher für Aufgabenstellungen wie z.B. Klassifikation an, bei denen die Ausgabe bekannt ist. Die zweite Möglichkeit ist für online lernende Modelle geeignet. Sie ist jedoch von einer Explosion der Komplexität bedroht, wenn viele Variablen und Fuzzy-Mengen im System verwendet werden müssen (bei zwei Eingangs- und einer Stellgröße mit je acht Fuzzy-Mengen gibt es 512, bei vier Eingangsgrößen bereits 32768 mögliche Regeln). Ein vollständiges Erlernen der Regelbasis ist daher auf diese Weise nur für wenig umfangreiche Aufgaben praktikabel. Meist läßt sich eine Regelbasis jedoch zumindest partiell angeben, so daß nur ein Teil der Regeln hinzugelernt werden muß.

Eine Lösung unseres Eingangsproblems, in dem ein hybrides Neuro-Fuzzy-System die Regelung einer Heizung übernimmt, könnte folgendermaßen aussehen: Zunächst legen Sie durch die Angabe von Fuzzy-Mengen fest, welche Temperaturen Sie als *angenehm* empfinden und welche nicht, zum Beispiel durch eine Einteilung der Temperaturskala in die durch geeignete Fuzzy-Mengen repräsentierte linguistischen Werte *zu niedrig, angenehm* und *zu hoch*. Anschließend bekommt das System den Auftrag, die Temperatur im *angenehmen* Bereich zu halten, und Sie übergeben ihm die Kontrolle über das Heizungsventil.

Da das System bisher noch keinerlei Kenntnis über die korrekte Vorgehensweise hat, müssen Sie zunächst eine etwas „chaotische“ Heizperiode überstehen, in der das Neuro-Fuzzy-System herauszufinden versucht, welche Fuzzy-Regeln eine *angenehme Temperatur* ergeben. Es überprüft, zu welchem Grad die aktuelle Temperatur Ihrer Vorstellung von *angenehm* entspricht. Wenn keine ausreichend hohe Übereinstimmung besteht, weiß das System, daß es sich in einem Fehlerzustand befindet, und bewertet diejenigen Regeln als *schlecht*, die zu diesem Zustand geführt haben. Entspricht die aktuelle Temperatur dagegen einer *angenehmen* Temperatur, so werden die verantwortlichen Regeln als *gut* bewertet.

Nach einer Weile hat das System durch Probieren und Bewerten selbständig eine Regelbasis gefunden und kann nun noch eine Optimierung der Regelung durch die Anpassung der

Fuzzy-Mengen vornehmen, die vom System zur Beschreibung der Temperatur verwendet werden. Dies geschieht nach derselben Vorgehensweise: Wenn ein Fehlerzustand erkannt wird, werden die Fuzzy-Mengen so verändert, daß beim erneuten Eintreten derselben Temperaturbedingungen eine Ventileinstellung erfolgt, die zu einer *angenehmeren* Temperatur führt. Abschließend können Sie sich die Fuzzy-Regeln und -Mengen anzeigen lassen und auf ihre Plausibilität hin überprüfen.

Sie haben somit nicht nur eine Problemlösung erhalten, sondern bekommen zusätzlich noch das Wissen vermittelt, wie das Problem zu lösen ist. Das Neuro-Fuzzy-System dient also auch dem Wissenserwerb, einer wichtigen Eigenschaft sogenannter „intelligenter Systeme“. Es entwickelt ähnlich wie ein Mensch nur durch Beobachtung des Prozeßverhaltens und unter Vorgabe eines anzustrebenden Prozeßzustandes eine geeignete Vorgehensweise. Natürlich läßt sich die erzielte Lösung oft weiter verbessern, wenn auf der Basis des vollständigen Verständnisses des Problems eine formale Modellierung gelingt (beispielsweise durch nichtlineare Differentialgleichungen) und geeignete formale Lösungsmethoden zur Verfügung stehen. Bei innovativen Produkten ist dies meistens gerade nicht der Fall, und häufig ist die von einem einfachen Neuro-Fuzzy-System gefundene Lösung bereits so gut, daß sich aus Sicht der Industrie der oft immense Aufwand einer optimalen Problemlösung nicht lohnt.

Auch in einer typischen Domäne Neuronaler Netze, der Klassifikation von Daten, lassen sich Neuro-Fuzzy-Systeme einsetzen. Wenn man einen guten statistischen oder neuronalen Klassifikator gefunden hat, so hat man in der Regel noch keine Information darüber erhalten, welche Zusammenhänge in den Daten die Klassifikation beeinflussen. Ein Klassifikator auf der Basis von Fuzzy-Regeln kann Auskunft über dieser Zusammenhänge geben. Es geht hier um die Frage, wie Fuzzy-Regeln aus Daten gewonnen werden können. Ein Ansatz stellt das sogenannte *Fuzzy Clustering* dar [Bezdek und Pal, 1992], das ähnliche Daten in gleiche Cluster sortiert. Die dabei gewonnen Regeln sind jedoch häufig schwer zu interpretieren, da sie bei ihre Bildung keiner Einschränkung unterliegen.

Das NEFCLASS-Modell (NEuro Fuzzy CLASSifcation, s. Abb. 4) geht einen anderen Weg [Nauck und Kruse, 1995, Nauck et al., 1996]. Das Modell ist ebenfalls von dem Fuzzy-Perceptron abgeleitet, verzichtet aber auf Fuzzy-Gewichte in der zweiten Verbindungsebene. Die in dem Modell eingesetzten Fuzzy-Regeln sind von der Form: *If* x_1 *is* A_1 *and* ... *and* x_n *is* A_n*, then class* C_1. Dabei sind die A_i linguitische Terme. Die Konklusion der Regel besteht nur in einer Klassenzuweisung. NEFCLASS lernt eine Regelbasis durch sukzessives Hinzufügen von Regeln, d.h. Knoten der inneren Schicht. Anschließend werden die verwendeten Fuzzy-Mengen optimiert. Idealerweise erhält man so einen Klassifikator, der in Form linguistischer Regeln interpretiert werden kann. Da bei der Bildung der Regeln und der Optimierung der Fuzzy-Mengen Einschränkungen definiert werden können, erhält man Regeln, die besser zu interpretieren sind, als die durch Fuzzy-Clustering gewonnenen Regeln. NEFCLASS verwendet ein überwachtes Lernverfahren und benötigt eine feste Lernaufgabe (Paare von Ein-/Ausgaben).

4 Zusammenfassung

In Japan gibt es bereits eine Vielzahl erfolgreicher Anwendungen Neuronaler Fuzzy-Systeme in den verschiedensten Gebieten. Man findet sie in Waschmaschinen, Kopiergeräten oder Systemen zur Bewertung von Börsenpapieren. Die Begriffe „Fuzzy-Logik“ und „Neuro“ werden in Japan als Werbeargumente benutzt. Dabei ist wie bereits erwähnt daß das Wort

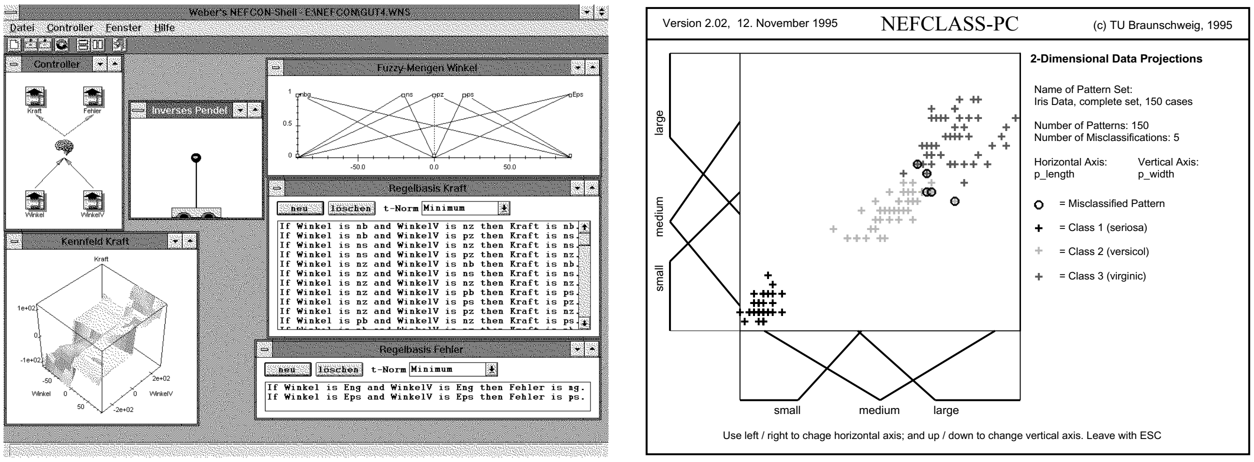

Abbildung 5: Die Oberflächen von NEFCON-Win und NEFCLASS-PC, zwei Implementierungen der angesprochenen Neuro-Fuzzy-Modelle für Personal Computer

„Logik" hier in einem sehr weiten, umgangssprachlichen Sinne zu verstehen. Methoden der mathematischen mehrwertigen Logik werden in den seltensten Fällen verwendet.

Aber auch deutsche Unternehmen befassen sich inzwischen mit Neuro-Fuzzy-Methoden. In [Steinmüller, 1994] wird eine Anwendung zur Regelung des Wasserverbrauchs einer Waschmaschine geschildert. Eine experimentelle Anwendung des NEFCON-Modells wird in [Eikens et al., 1995] diskutiert. Die Firma Siemens untersucht Neuro-Fuzzy-Modelle für Finanzanalysen und zur Modellierung des Deutschen Aktienindexes (DAX) [Zimmermann und Neuneier, 1996, Zimmermann et al., 1996].

Weitere Informationen und Software zu Neuro-Fuzzy-Systemen erhalten Sie auf unserer Homepage unter http://www.cs.tu-bs.de/ibr/fuzzy. Dort finden Sie auch Verweise auf Implementierungen von NEFCON und NEFCLASS (s. Abb. 5). Die Software ist in PC- und Unix-Versionen kostenlos erhältlich und dient zur Veranschaulichung des Lernverhaltens von Neuro-Fuzzy-Systemen.

Literatur

[Bezdek und Pal, 1992] J. C. Bezdek und S. K. Pal, Hrsg. (1992). *Fuzzy Models for Pattern Recognition.* IEEE Press, New York.

[Eikens et al., 1995] B. Eikens, V. Saucedo, M. Karim und A. Morris (1995). *Waste Water Neutralization Using a Fuzzy Neural Network Controller.* In *Proc. American Control Conference 1995.*

[Ichihashi, 1991] H. Ichihashi (1991). *Iterative Fuzzy Modelling and a Hierarchical Network.* In R. Lowen und M. Roubens, Hrsg.: *Proc. 4th IFSA Congress*, Band Engineering, Seiten 49–52, Brussels.

[Kruse et al., 1994a] R. Kruse, J. Gebhardt und F. Klawonn (1994a). *Foundations of Fuzzy Systems.* Wiley, Chichester.

[Kruse et al., 1995a] R. Kruse, J. Gebhardt und F. Klawonn (1995a). *Fuzzy–Systeme, 2. erweiterte Auflage.* Teubner, Stuttgart.

[Kruse et al., 1994b] R. Kruse, J. Gebhardt und R. Palm, Hrsg. (1994b). *Fuzzy Systems in Computer Science*. Vieweg, Braunschweig.

[Kruse et al., 1995b] R. Kruse, D. Nauck und F. Klawonn (1995b). *Neuronale Fuzzy–Systeme*. Spektrum der Wissenschaft, Juni(Heft 6/1995):34–41.

[Nauck, 1994] D. Nauck (1994). *Building Neural Fuzzy Controllers with NEFCON–I*. In [Kruse et al., 1994b], Seiten 141–151. Vieweg, Braunschweig.

[Nauck et al., 1996] D. Nauck, F. Klawonn und R. Kruse (1996). *Neuronale Netze und Fuzzy–Systeme, 2. erweiterte Auflage*. Vieweg, Wiesbaden.

[Nauck und Kruse, 1994] D. Nauck und R. Kruse (1994). *NEFCON–I: An X–Window based Simulator for Neural Fuzzy Controllers*. In *Proc. IEEE Int. Conf. Neural Networks 1994 at IEEE WCCI'94*, Seiten 1638–1643, Orlando.

[Nauck und Kruse, 1995] D. Nauck und R. Kruse (1995). *NEFCLASS – A Neuro–Fuzzy Approach for the Classification of Data*. In K. George, J. H. Carrol, E. Deaton, D. Oppenheim und J. Hightower, Hrsg.: *Applied Computing 1995. Proc. of the 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26–28*, Seiten 461–465. ACM Press, New York.

[Nomura et al., 1992] H. Nomura, I. Hayashi und N. Wakami (1992). *A Learning Method of Fuzzy Inference Rules by Descent Method*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, Seiten 203–210, San Diego.

[Pedrycz und Card, 1992] W. Pedrycz und H. C. Card (1992). *Linguistic Interpretation of Self-Organizing Maps*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, Seiten 371–378, San Diego.

[Rojas, 1993] R. Rojas (1993). *Theorie der Neuronalen Netze: Eine systematische Einführung*. Springer–Verlag, Berlin.

[Steinmüller, 1994] H. Steinmüller (1994). *Fuzzy Control und NeuroFuzzy bei Waschautomaten*. In H.-J. Zimmermann und C. von Altrock, Hrsg.: *Fuzzy Logic. Band2: Anwendungen*, Seiten 155–168. Oldenbourg Verlag, München.

[Zadeh, 1965] L. A. Zadeh (1965). *Fuzzy Sets*. Information and Control, 8:338–353.

[Zimmermann und Neuneier, 1996] H. G. Zimmermann und R. Neuneier (1996). *Neuro–Fuzzy Architectures And Learning Algorithms For Financial Analysis*. In *Proc. Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT96)*, Aachen.

[Zimmermann et al., 1996] H. G. Zimmermann, R. Neuneier, H. Dichtl und S. Siekmann (1996). *Modeling the German Stock Index DAX With Neuro–Fuzzy*. In *Proc. Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT96)*, Aachen.