

## 7 Neuro-Fuzzy in der Prozeßautomatisierung

T. Martinetz, J. Hollatz

---

Fuzzy-Logik besitzt gegenüber der klassischen Logik die Fähigkeit, unscharfe Information zu verarbeiten, und liefert somit für viele Bereiche, die mit numerischen Größen arbeiten, neue Lösungsansätze. Große Bedeutung kommt in jüngster Zeit den Untersuchungen zu, Fuzzy-Regelbasen durch Lernverfahren, wie mit neuronalen Netzen, zu generieren, um so unbekannte oder unvollständige Zusammenhänge erfassen zu können. Der folgende Beitrag führt in die Idee von Neuro-Fuzzy ein, zeigt die Äquivalenz von Fuzzy-Systemen und neuronalen Netzen mit radialen Basisfunktionen, und demonstriert das Konzept an einem Beispiel aus der Prozeßautomatisierung in der Walzwerktechnik.

### 7.1 Einleitung

Neuronale Netze und Fuzzy-Logik haben sich in den letzten Jahren auf den Gebieten der Informationsverarbeitung, Regelung und Steuerung als Schlüsseltechnologien fest etablieren können. Beide Ansätze sind fähig, hochdimensionale, nichtlineare Zusammenhänge nach menschlichem Vorbild zu modellieren. So arbeiten künstliche neuronale Netze mit sehr vielen zusammenspielenden Einheiten, die jeweils einfache Prozesse ausführen können. Durch die entstehende Gesamtdynamik der verbundenen Komponenten und die Fähigkeit, zu lernen, können komplexe Klassifikations- und Regressionsaufgaben gelöst werden. Diese subsymbolische Informationsverarbeitung ist der Biologie und Physiologie natürlicher Neuronennetze nachempfunden. Fuzzy-Logik dagegen resultiert aus dem Versuch, ein formales Modell menschlicher Logik aufzustellen. Durch die in den Regeln verwendeten unscharfen Begriffe können Lösungen für neue Situationen hergeleitet werden. Auch Abstraktionen und das damit verbundene Denken in Analogien ist erst durch die Flexibilität einer derartigen Logik möglich.

Neben neuronalen Netzen und Fuzzy-Logik hat sich seit geraumer Zeit auch Neuro-Fuzzy zu einem eigenständigen Gebiet entwickelt [1]. Dabei liegt die Idee dieser Kombination in der wechselseitigen Überführung von neuronalen Netzen in Fuzzy-Systeme und umgekehrt, um die Vorteile beider Methoden gemeinsam nutzen zu können. So ist es einem Neuro-Fuzzy-System möglich, durch ein in der Zielfunktion festgelegtes Kriterium und einen Adaptionalgorithmus die Parameter anzupassen und zu lernen. Zusätzlich zu diesen neuronalen Eigenschaften kann danach das in dem System modellierte Wissen in regelbasierter Form extrahiert werden. Somit ergibt sich die Möglichkeit, neu generierte bzw. optimierte Fuzzy-Regelsätze zu erhalten.

In dieser Arbeit wird ein Neuro-Fuzzy-Konzept vorgestellt und dessen Realisierung an einem Beispiel aus der Prozeßautomatisierung demonstriert. Dabei wird die Äquivalenz zwischen einem neuronalen Netz mit radialen Basisfunktionen und einem Fuzzy-System nach Takagi und Sugeno [2] gezeigt, wie auch die Frage nach optimalen Fuzzy-Regelbasen erörtert. Im letzten Teil werden Ergebnisse von einer Anwendung im Bereich der Walzwerksteuerung präsentiert.

## 7.2 Neuro-Fuzzy mit radialen Basisfunktionen

Um die Lernfähigkeit neuronaler Netze und die Transparenz von Fuzzy-Systemen zu nutzen, liegt es nahe, ein Konzept zu erarbeiten, das die Kombination beider Ansätze erlaubt. Es ist weiterhin wünschenswert, die beiden Methoden nicht nur nebeneinander kombinieren zu können, sondern gleichzeitig deren wechselseitige Transformation zu ermöglichen, ohne daß sie ihre eigenen elementaren Eigenschaften verlieren. Im Sinne dieser Forderungen ist folgendes Neuro-Fuzzy-Modell entstanden [3, 4].

Wir betrachten ein neuronales Netz  $y = NN(x)$ , das ein Ergebnis  $y \in \mathfrak{R}$  für eine gegebene Eingabe  $x \in \mathfrak{R}^n$  errechnet. Zusätzlich gehen wir davon aus, daß ein Experte Informationen in Form eines Regelsatzes über die gleiche Abbildung bereitstellt. Die Prämisse einer Regel spezifiziert die Bedingungen an  $x$ , unter denen die Folgerung eintritt. Der Bereich des Eingaberaums, in dem die Bedingung zutrifft, wird formal durch eine Basisfunktion  $b_i(x)$  beschrieben. Anstatt nur binäre Zustände für die Basisfunktionen zuzulassen (1: Prämisse trifft zu, 0: Prämisse trifft nicht zu), verwenden wir kontinuierliche positive Werte, welche die Sicherheit oder Gewichtung der Regel bei gegebener Eingabe beschreiben.

Wir nehmen an, daß die Folgerung einer Regel in Form eines mathematischen Ausdrucks beschrieben werden kann, so wie: *Folgerung: die Ausgabe ist gleich  $w_i(x)$* , wobei  $w_i(x)$  eine Funktion der Eingabe ist. Dabei kann der Ausdruck  $w_i(x)$  eine Konstante, ein Polynom oder ein anderes neuronales Netz sein.

Wenn die Bedingungen mehrerer Regeln bei gegebenen Eingabewerten zutreffend sind, definieren wir die auszugebende Schlußfolgerung aller Regeln als den gewichteten Durchschnitt der Ausgaben der Regeln mit erfüllten Bedingungen. Dabei ist der Gewichtungsfaktor bei gegebener Eingabe proportional der Basisfunktionsaktivität, und wir erhalten als Netzausgabe

$$y = NN(x) = \frac{\sum_i w_i(x) b_i(x)}{\sum_j b_j(x)}. \quad (1)$$

Dies ist ein sehr allgemeines Konzept, da man völlig frei ist, die Form der Basisfunktionen  $b_i(x)$  und die Folgerungen  $w_i(x)$  zu spezifizieren. Können die Basisfunktionen mit einer multivariaten Gaußfunktion approximiert werden, und sind die  $w_i$  Konstanten, so erhält man ein Netzwerk mit normalisierten Basisfunktionen, das schon von Moddy und Darken [5] verwendet wurde.

### 7.2.1 Netzwerke aus radialen Basisfunktionen

Ein Netzwerk aus radialen Basisfunktionen (RBF-Netz) ist ein zweischichtiges Netzwerk, dessen Ausgabeknoten eine Linearkombination von Basisfunktionen berechnet. Diese Basisfunktionen sind in den Knoten der versteckten Schicht repräsentiert und produzieren eine lokale Antwort auf die Eingangsgrößen. Typischerweise gibt ein solcher Knoten nur dann eine signifikant von Null verschiedene Antwort, wenn die Eingabe in einen kleinen lokalen Bereich des Eingaberaumes fällt. Aus diesem Grund wird diese neuronale Struktur auch als Netz mit lokalen rezeptiven Feldern (*localized receptive field network*) bezeichnet. Die einzeln veränderbaren und gegenseitig überlappenden rezeptiven Felder sind Strukturen, wie man sie in Regionen des zerebralen und des visuellen Kortex findet. Obwohl es viele Möglichkeiten der Wahl von Basisfunktionen gibt, ist die am häufigsten verwendete Basis eine multivariate Gaußsche Funktion der Form

$$b_i(\mathbf{x}) = \kappa_i \cdot \exp\left[-\sum_{j=1}^L \frac{1}{2\sigma_{ij}^2}(x_j - c_{ij})^2\right] \quad i = 1, \dots, N, \quad (2)$$

wobei  $b_i(\mathbf{x})$  die Ausgabe des  $i$ -ten Knotens in der versteckten Schicht ist,  $\kappa_i$  ein positiver, reeller Gewichtungswert der Basisfunktion,  $\mathbf{x} = (x_1, \dots, x_L)$  das  $L$ -dimensionale Eingabemuster und  $c_{ij}$  die Gewichte des  $i$ -ten Knotens in der versteckten Schicht, welche mit  $\mathbf{c}_i = (c_{i1}, \dots, c_{iL})$  das Zentrum der  $i$ -ten Gaußschen Basisfunktion spezifizieren.  $\sigma_{ij}$  ist die Weite der  $i$ -ten Basisfunktion entlang der  $j$ -ten Eingangsdimension, und  $N$  ist die Anzahl der Knoten in der versteckten Schicht. Je näher die Eingabe am Zentrum der Gaußschen Funktion liegt, umso größer ist die Ausgabe des Knotens. Der Name *radiale Basisfunktion* resultiert daher, daß die Gaußsche Glockenkurve, falls  $\sigma_{ij} = \sigma_i$  gilt, radial symmetrisch ist. So generiert ein Knoten identische Ausgaben für alle Eingaben, die in einer bestimmten radialen Entfernung vom Zentrum  $\mathbf{c}_i$  liegen.

### 7.2.2 Äquivalenz von RBF-Netzen und Fuzzy-Inferenzsystemen

Die funktionelle Gleichheit zwischen RBF-Netzen und Fuzzy-Inferenz-Systemen von Takagi und Sugeno [2] kann gezeigt werden, wenn folgende Punkte zutreffen:

1. Die Anzahl der Basisfunktionen (rezeptiven Felder, versteckten Knoten) ist gleich der Anzahl der Fuzzy-wenn-dann-Regeln.
2. Die Ausgabe jeder Fuzzy-wenn-dann-Regel kann entweder eine Konstante im einfachen Fall, oder aber auch eine Funktion abhängig vom Eingangsvektor sein.
3. Die Zugehörigkeitsfunktionen jeder einzelnen Regel werden als Gaußsche Funktionen gewählt.
4. Der T-Norm-Operator, der benutzt wird um die Gültigkeit jeder einzelnen Regel zu berechnen, ist die Multiplikation.

5. Sowohl das RBF-Netz als auch das Fuzzy-Inferenz-System benutzen unter den genannten Voraussetzungen die gleiche Methode (gewichtete Summe oder gewichtetes Mittel) zur Berechnung der Gesamtausgabe.

Wie oben beschrieben können mit diesen Netzen Fuzzy-Regelsysteme dargestellt werden. Die Transformationsmethode kann an folgendem Beispiel demonstriert werden. Wenn die linguistischen Variablen durch das Zentrum ihrer Zugehörigkeitsfunktion repräsentiert sind, könnte eine Regel z.B. folgendermaßen aussehen:

**Wenn**  $[(x_1 = A) \text{ und } (x_2 = B)] \text{ oder } (x_3 = C)$  **dann**  $y = d \times x^2$ .

Dabei sind A,B und C linguistische Variablen, die Fuzzy-Mengen repräsentieren, und deren Gaußsche Zugehörigkeitsfunktionen die Mittelwerte a,b und c haben. Diese Fuzzy-Regel wird im Netz codiert mit

$$\text{Bedingung}_i : b_i(\mathbf{x}) = \exp\left[-\frac{1}{2} \frac{(x_1 - a)^2 + (x_2 - b)^2}{\sigma^2}\right] + \exp\left[-\frac{1}{2} \frac{(x_3 - c)^2}{\sigma^2}\right]$$

$$\text{Folgerung}_i : w_i(\mathbf{x}) = d \times x^2.$$

Durch die Gleichartigkeit der Zielsetzung beider Darstellungsformen kann Regelextraktion in neuronale Netze und Lernen in Fuzzy-Systeme eingebracht werden. Um die Aussagekraft der gelernten und extrahierten Fuzzy-Regeln zu optimieren, sind noch folgende Überlegungen zu machen:

In der Regel läuft der Adaptionsprozeß eines neuronalen Netzes in einem dreistufigen Prozeß ab. Ein Netz wird mit Hilfe der Trainingsdaten adaptiert. Anschließend wird der Fehler auf einer Kreuzvalidierungsmenge berechnet, der eine Abschätzung des Generalisierungsfehlers darstellt. Darauf folgt eine Modifikation der Komplexität des Netzes. Diese drei Schritte werden wiederholt ausgeführt, mit dem Ziel, den Kreuzvalidierungsfehler zu minimieren. Nach dem Training können Regeln wieder aus dem Netz extrahiert werden, wobei dafür zu sorgen ist, daß der gewonnene Regelsatz so kompakt und aussagekräftig wie möglich ist. Der Wert der extrahierten Regeln ist sonst doch in seiner Aussagekraft sehr beschränkt. Man möchte eine möglichst kleine Anzahl von Regeln haben, die das gelernte Wissen ausreichend beschreiben. Zusätzlich soll das Netz für jede Regel die kleinste Anzahl von Konjunktionen finden, d.h. jede Regel (Basisfunktion) soll von möglichst wenigen Eingangskomponenten abhängig sein. Um dies zu erreichen, wird folgendes Vorgehen für Gaußsche Basisfunktionen vorgeschlagen:

1. *Eliminiere Basisfunktionen.* Berechne mit  $\tau_i = b_i(c_i) \sum_j b_j(c_j)$  das relative Gewicht jeder Basisfunktion an der Stelle ihres Zentrums. Diese Größe stellt ein Wichtigkeitsmaß der Basisfunktion für das Netz dar. Eliminiere den Knoten  $i$  mit dem kleinsten Wert des Wichtigkeitsmaßes  $\tau_i$ .
2. *Eliminiere Konjunktionen.* Setze den größten  $\sigma_j$ -Wert auf unendlich. Dadurch wird im Endeffekt die Eingabe  $j$  in der Basisfunktion  $i$  gelöscht.

Eliminiere schrittweise Basisfunktionen und Konjunktionen bis der Fehler über einen Schwellwert steigt. Nach jedem Eliminationsschritt wird das Netz wieder trainiert.

Die einfache Konzeption dieses Neuro-Fuzzy-Ansatzes ermöglicht den Einsatz in einer großen Problempalette. Im nächsten Abschnitt wird der Einsatz an einem Beispiel aus der Prozeßautomatisierung beschrieben.

### 7.3 Bestimmung von Walzkraften mit Neuro-Fuzzy

Im folgenden möchten wir an einer Problemstellung aus dem Bereich der Prozeßautomatisierung in der Walzwerktechnik, der Walzkraftberechnung an einer Warmbreitbandstraße, die Anwendung der oben beschriebenen Neuro-Fuzzy Methode demonstrieren. Das zentrale Problem beim Warmwalzen von Stahlbändern ist die Berechnung der Walzkraft zur Erzielung der gewünschten Banddickenabnahme an dem betrachteten Walzgerüst. Die erforderliche Walzkraft hängt dabei von einer Reihe von Einflußgrößen ab. Die wichtigsten sind in Bild 7.1 aufgeführt. Neben der gewünschten relativen Banddickenabnahme  $\varepsilon$  hängt die benötigte Walzkraft von der Dicke  $d$  des Bandes beim Eintritt in das Walzgerüst, seiner Breite  $b$  sowie seiner Temperatur  $T$ , welche maßgeblich die Plastizität des Materials bestimmt, ab. Zusätzlich zur Temperatur muß als Einflußgröße auf die Plastizität des Bandes aber auch noch dessen chemische Zusammensetzung berücksichtigt werden, wobei sich bei Auswertung der Meßdaten von der Walzstraße insbesondere die Konzentration von Vanadium  $V$  als signifikant heraus-

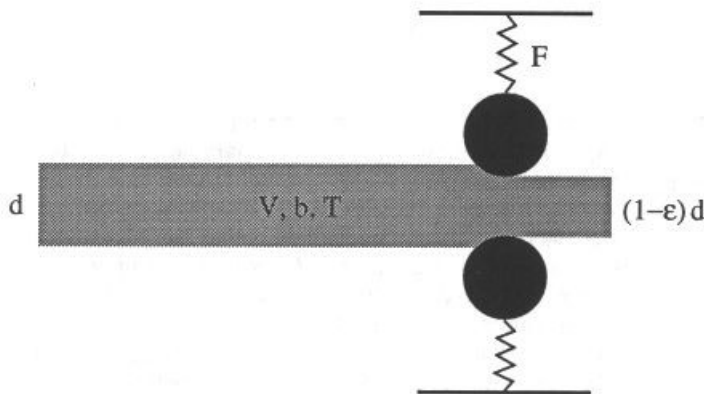


Bild 7.1: Die an einem Walzgerüst einer Warmbreitbandstraße einzustellende Walzkraft  $F$  wird wesentlich von den folgenden fünf Einflußfaktoren bestimmt: Eintrittsdicke  $d$ , Vanadiumkonzentration  $V$ , Bandtemperatur  $T$ , Bandbreite  $b$  sowie relative Banddickenreduktion  $\varepsilon$

stellte. Die Aufgabenstellung lautet nun, auf der Basis der hier aufgeführten fünf wichtigsten Einflußfaktoren  $V, b, T, d$  und  $\varepsilon$  einen Fuzzy-Regelsatz aufzustellen, mithilfe dessen es möglich ist, die erforderliche Walzkraft zur Erzielung der gewünschten Banddickenabnahme abzuleiten.

Zur Erstellung des Fuzzy-Regelsatzes mit der oben beschriebenen Neuro-Fuzzy Methode müssen wir ein RBF-Netzwerk spezifizieren, welches wir anschließend mit Meßdaten von der Walzstraße trainieren. Mit zehn Gaußförmigen Basisfunktionen wählen wir eine anfängliche Netzwerkgröße, welche mehr Regeln beschreiben kann als bei der betrachteten Problemstellung zu erwarten ist. Die Folgerungen  $w_i(x)$  wählen wir als Konstanten  $w_i$ . Die Meßdaten von der Walzstraße bestehen aus  $(x, F)$  Datenpaaren, mit  $x$  als fünfdimensionalem Eingangsvektor, welcher die oben spezifizierten fünf Einflußgrößen als Komponenten enthält, und  $F$  als der bei dem gemessenen Band tatsächlich erforderlich gewesenene Walzkraft. Insgesamt stehen uns Datenpaare von 10000 Bändern zur Verfügung.

Das Training des RBF-Netzwerkes, das Adaptieren seiner Parameter  $w_i, x_i, c_i, \sigma_i$ , geschieht über Gradientenabstieg auf einer Fehlerfunktion  $E$ . Die Fehlerfunktion  $E$  ist die Summe der quadratischen Fehler der Netzwerkausgabe  $NN(x)$  über die 10000 Trainingsdaten  $(x_n, F_n)$ ,  $n = 1, \dots, 10000$  und ist gegeben durch

$$E = \sum_{n=1}^{10000} (F_n - NN(x_n))^2. \quad (3)$$

Die Ableitung von  $E$  nach dem entsprechenden Netzwerkparameter, zum Beispiel  $w_i$ , liefert die Adaptiongleichung für diesen Netzwerkparameter, für  $w_i$  die Gleichung

$$\Delta w_i = \eta \sum_{n=1}^{10000} (F_n - NN(x_n)) \frac{\partial NN(x_n)}{\partial w_i}. \quad (4)$$

Setzt man für  $NN(x_n)$  Gleichung (1) und (2) ein, so erhält man die explizite mathematische Form der entsprechenden Lernregel. Der Lernparameter  $\eta$  bestimmt die Schrittweite des Gradientenabstiegs.

Nach der Trainingsphase, nachdem durch Gradientenabstieg das Minimum auf der Fehlerfunktion  $E$  gefunden wurde, ergeben sich für die Parameter des RBF-Netzes die in Tabelle 7.1 aufgelisteten Werte. Insgesamt erhalten wir zehn Fuzzy-Regeln, entsprechend der Anzahl der beim RBF-Netz eingesetzten Basisfunktionen. Die Mehrzahl der Regeln ist jedoch von geringer Relevanz, wie sich aus den Werten für  $\tau_i$  ersehen läßt. Sehen wir nur diejenigen Regeln (Basisfunktionen) als relevant an, deren  $\tau_i$  mindestens noch 20 % vom höchsten  $\tau$ -Wert annehmen, so reduziert sich die Zahl der zu berücksichtigenden Regeln auf vier. Übrig bleiben Regel 1,2,3 und Regel 7. Durch die Reduktion der Anzahl der Regeln wird deren Aussagekraft beträchtlich erhöht, jedoch muß berücksichtigt werden, daß dadurch die Ausgabegenauigkeit des Systems etwas sinken kann.

	$Bf_1$	$Bf_2$	$Bf_3$	$Bf_4$	$Bf_5$	$Bf_6$	$Bf_7$	$Bf_8$	$Bf_9$	$Bf_{10}$
$\tau_i$	5.4	20.7	4.5	1.2	2.4	1.0	15.1	2.9	2.6	3.6
$1/\sigma_{ij}$										
V	0.1	1.5	0.3	0.4	0.5	0.4	0.0	0.2	0.2	0.0
b	0.4	0.4	0.0	0.0	0.7	0.0	0.3	1.2	0.0	0.1
T	0.8	0.0	0.1	0.7	0.3	0.0	0.9	0.1	0.1	1.2
d	0.0	0.0	1.8	0.2	0.5	0.6	0.0	0.0	0.0	0.6
$\epsilon$	1.1	0.7	0.0	0.4	0.7	0.0	0.9	0.2	0.0	0.0
$c_{ij}$										
V	0.1	0.1	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1	-0.0	5.2
b	-0.7	0.9	0.9	-1.4	-0.9	0.4	0.6	-0.5	1.7	-1.4
T	1.1	-1.7	-0.3	-0.2	-0.7	1.2	-1.3	-0.1	0.6	-0.6
d	0.4	0.1	1.4	-0.6	1.2	-0.2	-0.5	-0.7	-0.2	5.0
$\epsilon$	-0.1	2.0	-1.2	0.8	-1.1	0.2	1.2	0.4	-0.2	-2.7
$w_i[MN]$	-28.9	14.7	-6.8	-58.2	-36.3	-87.9	12.6	3.9	9.8	-41.2

Tabelle 7.1: Die Werte der Netzwerkparameter nach dem Training. Jede Spalte enthält die Parameter einer Basisfunktion ( $Bf_i$ ). Nur die eingerahmten Spalten haben ausreichend hohe  $\tau$ -Werte und beschreiben damit relevante Basisfunktionen. Die zunächst zehn Basisfunktionen (Regeln) reduzieren sich somit auf vier

Als nächsten Schritt wollen wir die Eingangsgrößen und damit die Konjunktionen der verbleibenden vier Regeln reduzieren. Dies geschieht durch Auswertung der Parameter  $\sigma_{ij}$ . Ist der  $\sigma_{ij}$ -Wert einer Eingangsgröße  $x_j$  groß ( $1/\sigma_{ij}$  klein), so bedeutet dies, daß bei einer Änderung der Eingangsgröße  $x_j$  der Ausgabewert der betrachteten  $i$ -ten Basisfunktion (Regel) und damit der Ausgabewert des gesamten RBF-Netzes sich nur wenig ändert. Dann jedoch ist der Ausgabewert der betrachteten Basisfunktion praktisch unabhängig vom Wert dieser Eingangsgröße, und diese braucht somit bei der entsprechenden Regel nicht berücksichtigt werden. Auch hier gelten wieder die von uns festgelegten 20 % vom höchsten  $1/\sigma_{ij}$ -Wert als Minimum, um als relevant anerkannt zu werden.

Nach Reduktion der Eingangsgrößen anhand der  $\sigma$ -Werte gehen nun in die Prämisse von Regel 1 die Breite  $b$ , die Bandtemperatur  $T$  und die relative Dickenabnahme  $\epsilon$  ein, in die Prämisse von Regel 2 die Größen  $V$ ,  $b$  und  $\epsilon$ , in die Prämisse von Regel 3 nur die Größe  $d$ , und in die Prämisse der vierten relevanten Regel, Regel 7, die Größen  $b$ ,  $T$  und  $\epsilon$ . Die Schlußfolgerung bei Gültigkeit der Prämisse einer Regel ist dann eine Korrektur der Walzkraft um einen bestimmten Betrag, bei Gültigkeit der Prämisse von Regel  $i$  um den Betrag  $w_i$ .

Damit ist die Reduktion der Regeln und der Eingangsgrößen abgeschlossen, und um die Fuzzy-Regeln aufstellen zu können, müssen wir lediglich noch die Werte

für die Zentren  $c_{ij}$  der Basisfunktionen in linguistische Variable transformieren, welche die entsprechenden Fuzzy-Mengen und deren Zugehörigkeitsfunktionen repräsentieren. Wie oben bereits erläutert, geben die Zentren  $c_{ij}$  an, in welchem Eingangsraumbereich die entsprechende Regel Gültigkeit besitzt, womit die  $c_{ij}$  jeweils eine Zugehörigkeitsfunktion definieren. Um die Werte für die  $c_{ij}$  in linguistische Variablen transformieren zu können, haben wir jede Eingangsgröße  $x_j$ ,  $j = 1, \dots, 5$  derart skaliert, daß ihr mittlerer Wert  $\bar{x}_j$ , gemittelt über die Trainingsdaten, bei Null zu liegen kommt und die  $\bar{x}_j - \sigma_j$  und  $\bar{x}_j + \sigma_j$  Werte bei -1 bzw. +1, mit  $\sigma_j$  als die Varianz der j-ten Eingangsgröße (nicht zu verwechseln mit den Breiten  $\sigma_{ij}$  der radialen Basisfunktionen). Mit der Skalierung der Eingangsgrößen  $x_j$  werden auch entsprechend die  $c_{ij}$  Werte skaliert. In Tabelle 7.1 sind die bereits skalierten  $c_{ij}$  Werte aufgelistet.

Entscheiden wir uns für *niedrig*, *mittel* und *hoch* als die möglichen Werte der linguistischen Variablen, so ordnen wir einer Zugehörigkeitsfunktion die linguistische Variable niedrig zu, falls  $c_{ij} < -0.5$ , mittel falls  $-0.5 < c_{ij} < 0.5$  und hoch falls  $c_{ij} > 0.5$ .

Damit ergeben sich für das Problem der Bestimmung der Walzkraft folgende Fuzzy-Regeln:

1. Wenn  $b$  niedrig,  $T$  hoch und  $\varepsilon$  mittel, dann reduziere die Walzkraft um -28.9 MN.
2. Wenn  $V$  mittel,  $b$  hoch und  $\varepsilon$  hoch, dann erhöhe die Walzkraft um 14.7 MN.
3. Wenn  $d$  hoch, dann reduziere die Walzkraft um -6.8 MN.
4. Wenn  $b$  hoch,  $T$  niedrig und  $\varepsilon$  hoch, dann erhöhe die Walzkraft um 12.6 MN.

Dieser aus den Meßdaten abgeleitete Fuzzy-Regelsatz beschreibt ein Fuzzy-System nach Takagi und Sugeno [2] mit nicht fuzzyfzierter Ausgabe. Anzumerken ist, daß die Fuzzy-Mengen und ihre Zugehörigkeitsfunktionen bei den obigen Regeln, festgelegt durch die  $c_{ij}$ , im Gegensatz zu konventionellen Fuzzy-Regelsätzen nicht nur für jede Eingangsgröße  $x_j$ , sondern zusätzlich noch für jede Regel  $i$  verschieden sein können. Strenggenommen müßten die Zugehörigkeitsfunktionen *niedrig*, *mittel* und *hoch* daher den Index  $i$  tragen.

Die vier Regeln lassen sich jetzt interpretieren. Betrachten wir zunächst Regel 1: Diese besagt, daß schmale Stahlbänder mit hoher Temperatur eine verminderte Walzkraft erfordern. Dies macht Sinn, denn schmale Bänder weisen eine geringere Kontaktfläche mit der Walze auf als breite Bänder, und Bänder hoher Temperatur sind außerdem „weicher“ als Bänder mit niedriger Temperatur. Auch Regel 2 läßt sich interpretieren. Sie besagt unter anderem, daß bei breiten Bändern, bei denen außerdem eine große Dickenreduktion vorgenommen werden soll, die Walzkraft stark erhöht werden muß. Breite Bänder haben eine große Kontaktfläche, und bei großen Dickenabnahmen müssen natürlich entsprechend hohe



Walzkraften angewandt werden. Besonders interessant ist, daß sich eine Regel wie Nr.3 aus den Meßdaten ergibt. Diese besagt, daß bei Bändern mit einer großen Eintrittsdicke die Walzkraft leicht reduziert werden muß. Dieses Verhalten der Walzkraft wird tatsächlich an der Walzstraße beobachtet und resultiert daher, daß Bänder mit einer großen Eintrittsdicke bis dato erst eine geringe Stauchung erfahren haben und daher deren Materialgefüge noch großen „Spielraum“ für Banddickenreduktionen bietet. Im Vergleich zu bereits stark gestauchten Bändern mit einer entsprechend kleinen Eintrittsdicke wird daher, bei gleicher relativer Banddickenreduktion, eine geringere Walzkraft benötigt. Regel 4 ist im wesentlichen reziprok zu Regel 1. Bänder mit großer Breite und daher großer Kontaktfläche, niedriger Temperatur und daher geringer Plastizität sowie hoher relativer Banddickenreduktion erfordern erwartungsgemäß hohe Walzkraften.

Vergleicht man den mittleren Walzkraftfehler des Neuro-Fuzzy Verfahrens mit dem mittleren Walzkraftfehler bei konventioneller Vorgehensweise, so zeigt sich, daß bereits mit den obigen vier Regeln eine Genauigkeit erreicht wird, die annähernd im Bereich des bisherigen Verfahrens liegt. Das konventionelle Verfahren, basierend auf einer Reihe physikalischer Modelle, dessen Parameter permanent adaptiert werden, ist allerdings erheblich aufwendiger und weniger durchschaubar.

#### **7.4 Zusammenfassung**

Wir haben gezeigt, daß mit einem Neuronalen Netz vom Typ eines sogenannten RBF-Netzes Fuzzy-Inferenzsysteme nach Takagi und Sugeno [2] dargestellt werden können. Dies ermöglicht es, die wesentliche Eigenschaft Neuronaler Netze, Zusammenhänge adaptiv modellieren zu können, auszunutzen, um neue Fuzzy-Regelsätze basierend auf Meßdaten abzuleiten oder bereits existierende Regelsätze anhand dieser Meßdaten zu optimieren. Wichtig sind dabei Mechanismen zum Eliminieren überflüssiger Regeln sowie überflüssiger Eingangsgrößen. Ein geeignetes Verfahren wurde vorgestellt und erläutert.

An einer zentralen Problemstellung aus dem Bereich der Prozeßautomatisierung von Walzstraßen, der Bestimmung der Walzkraften, wurde die Vorgehensweise bei Anwendung des hier präsentierten Neuro-Fuzzy Konzeptes demonstriert. Es zeigt sich, daß bereits vier Regeln ausreichen, um annähernd die Walzkraftgenauigkeit des wesentlich aufwendigeren konventionellen Verfahrens zu erreichen.

## 7.5 Literatur

- [1] Tresp, V., Preuß, H.-P.: Neuro-Fuzzy, VDI Berichte Nr.113 (1994) 89–122.
- [2] Takagi, T., Sugeno, M.: *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, No. 1 (1985) 116–132.
- [3] Hollatz, J.: *Integration von regelbasiertem Wissen in neuronale Netze*, Dissertation, Institut für Informatik, Technische Universität München (1993).
- [4] Hollatz, J., Tresp, V.: *A Rule-based Network Architecture*, Proceedings of ICANN 92, Brighton (1992) 757–761.
- [5] Moody, J., Darken, C.: *Fast learning in networks of locally-tuned processing units*, Neural Computation, Vol. 1 (1989) 281–294.