

# Adaptive Resonance Theory (ART)

Sebastian Nusser ([nusser@mail.cs.uni-magdeburg.de](mailto:nusser@mail.cs.uni-magdeburg.de))

**Abstract.** Gail Carpenter und Stephen Grossberg entwickelten mit ART eine neue Familie Neuronaler Netze. Diese Familie ist in der Lage, das Stabilitäts-Plastizitäts-Dilemma Neuronaler Netze zu lösen. Das Grundprinzip dieser stark biologisch motivierten Architektur beruht auf dem Versuch, den Eingabevektor anhand eines festgelegten Ähnlichkeitsmaßes zu einer bereits bestehenden Kategorie zuzuordnen. Bei ausreichender Ähnlichkeit kann der entsprechende Referenzvektor in geringem Maße durch den Eingabevektor angepasst werden – die anderen Referenzvektoren werden nicht verändert. Besteht zu keinem bereits existierenden Referenzvektor eine ausreichende Ähnlichkeit, wird eine neue Kategorie mit dem Eingabevektor als Prototyp erzeugt.

## 1. Einführung

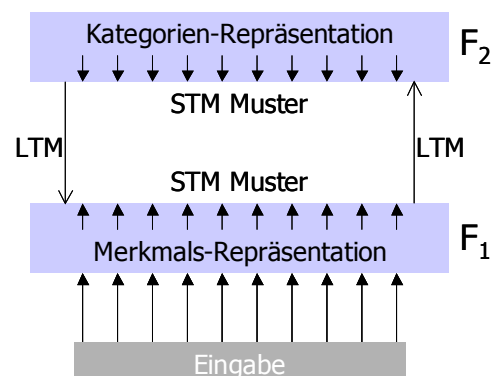
Bei dem Modell der Adaptive Resonance Theory (ART) handelt es sich um eine biologisch motivierte Familie Neuronaler Netze, die von Gail Carpenter und Stephen Grossberg (z.B. [Gross1976], [CarGro1987], [CarGro1990] und [CaGrRo1991]) entwickelt wurde. Dabei wird das Netz in die Lage versetzt, Informationen wie ein natürliches Nervensystem zu erfassen und zu ordnen. Das heißt, es kann in bereits gelerntes Wissen neues Wissen integrieren, ohne das es nötig ist, wieder alle alten Muster zuzüglich der neuen Muster zum erneuten Lernen zu verwenden. Es genügt die Präsentation eines neuen Musters zum Lernen – bereits korrekt gelernte Muster bleiben aber erhalten.

## Hintergrund

Das Modell der Adaptive Resonance Theory beruht auf drei grundlegenden Annahmen [Krösma1996]:

1. Normalisierung der Netzwerkaktivität: Lebewesen sind in der Lage, ihre Wahrnehmung an veränderte Umweltbedingungen (z.B. ein erhöhter Lärmpegel) anzupassen.
2. Kontraststeigerung: die Kenntnis von feinen Unterschieden in den Eingabemustern (z.B. die Unterscheidung bestimmter Emotionszustände des Gegenübers) kann entscheidend für eine situationsbedingte Reaktion sein.
3. Kurzzeit- und Langzeitgedächtnis: das Kurzzeitgedächtnis (STM) speichert die differenzierten Eingabemuster. Das Langzeitgedächtnis (LTM) dient der Klassifikation und kann durch das STM variiert werden.

Das Netz besteht aus zwei Schichten. Die Vergleichsschicht ( $F_1$ ) und die Erkennungsschicht ( $F_2$ ) sind über ein Langzeitgedächtnis miteinander verbunden. Das Eingabemuster wird in die  $F_1$ -Schicht übertragen – die  $F_2$ -Schicht klassifiziert das Muster. Die Eingabe beeinflusst die Aktivierung der Merkmals-



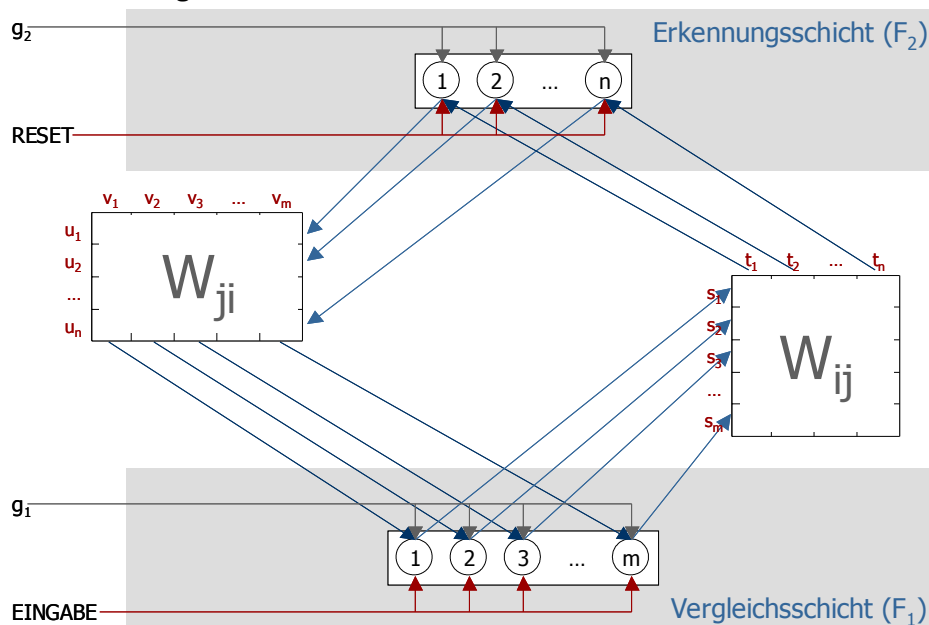
repräsentation. Über Verbindungen des Langzeitgedächtnisses wird die Eingabe zur Kategorienrepräsentation übertragen. Die dadurch entstandene Klassifikation wird über eine Gewichtung der Informationen des Langzeitgedächtnisses zurück in die  $F_1$ -Schicht übertragen. Hier wird verglichen, ob die Zuordnung mit der Eingabe übereinstimmt. Trifft dies zu wird die Gewichtung der Informationen verstärkt – sonst wird die Klassifikation abgelehnt.

## 2. Aufbau und Funktionsweise von ART-1

Bei ART-1 handelt es sich um die einfachste ART-Architektur, welche sich nur für die Verarbeitung binärer Eingabevektoren eignet. Sie bildet die Grundlage für weitere ART-Netze.

### Grundstruktur

Ein ART-1-Netz ist ein zweischichtiges, rückgekoppeltes Neuronales Netzwerk, bei dem jedes Neuron aus der Vergleichsschicht über eine reellwertige Gewichtsmatrix  $W_{ij}$  mit einem Neuron der Erkennungsschicht verbunden ist. Die Neuronen aus der Erkennungsschicht sind wiederum über eine binäre Matrix  $W_{ji}$  mit den Neuronen der Vergleichsschicht verbunden. Weiterhin besitzt das Netz zwei Verstärkungsfaktoren ( $g_1$  und  $g_2$ ), die der Synchronisation des Netzwerks dienen, und eine Reset-Komponente, die die Neuronen der Erkennungsschicht deaktivieren kann.



### Verstärkungsfaktoren und Reset-Komponente

Die Verstärkungsfaktoren dienen als Schalter zur Netzwerksynchronisation:

- $g_1$  ist aktiv, wenn mindestens eine Komponente des Eingabevektors  $I$  gleich 1 und keine Komponente der Erkennungsschicht gleich 1 ist ( $g_1 = (I_1 \vee I_2 \vee \dots \vee I_m) \wedge \neg(u_1 \vee u_2 \vee \dots \vee u_n)$ )
- $g_2$  ist aktiv, wenn kein Nullvektor als Eingabe vorliegt ( $g_2 = I_1 \vee I_2 \vee \dots \vee I_m$ )

Unterscheiden sich der Eingabevektor  $I$  und der Ausgabevektor der Vergleichsschicht  $S$  um mehr als einen Toleranzparameter  $\rho$ , wird die Reset-Komponente auf 1 gesetzt. Dadurch wird ein Feuern der Neuronen aus der Erkennungsschicht verhindert.

### Vergleichsschicht und 2/3-Regel

Für jedes Neuron  $i$  aus der Vergleichsschicht gibt es 3 unterschiedliche Eingaben:

- $I_i$  als Komponente des Eingabevektors
- $g_i$  als Verstärkungssignal (bei allen Neuronen gleich)
- $v_i$  als gewichtete Summe der Ausgaben der Erkennungsschicht

Um den Ausgang  $s_i$  eines Neurons  $i$  der  $F_1$ -Schicht zu aktivieren, werden zwei aktive Eingaben benötigt. Dies wird auch als 2/3-Regel bezeichnet:

$$s_i = \begin{cases} 1 & , \text{ falls } ((I_i \wedge v_i) \vee (I_i \wedge g_i) \vee (v_i \wedge g_i)) \\ 0 & , \text{ sonst} \end{cases}$$

Bei der Initialisierung des Netzes ist  $g_i = 1$  und es gilt für alle  $i$   $v_i = 0$ , somit ergibt sich  $s_i = I_i$ .

### *Erkennungsschicht*

In der Erkennungsschicht findet die Zuordnung der Eingabe in eine existierende bzw. in eine neue Klasse statt. Dies ist von der Ähnlichkeit des Eingabevektors zu den gespeicherten Mustern abhängig.

Nur das Neuron  $J^1$  wird aktiviert, dessen Gewichtsvektor  $W_j = (w_{1j}, w_{2j}, \dots, w_{mj})$  die größte Ähnlichkeit zu  $S$  besitzt. Ein Maß für diese Ähnlichkeit ist beispielsweise das Skalarprodukt:

$$u_j = \begin{cases} 1 & , \text{ falls } t_j = \sum_i s_i w_{ij} = \max \\ 0 & , \text{ sonst} \end{cases}$$

### **Arbeitsweise**

Nach der Initialisierung beinhaltet die Arbeitsweise eines ART-1-Netzwerkes die folgenden Phasen:

- Erkennungsphase (recognition)
- Vergleichsphase (comparison)
- Suchphase (search)
- Gewichtsadaption (training)

### *Initialisierung*

Man setzt alle Gewichte  $w_{ij}$  der reellwertigen Gewichtsmatrix  $W_{ij}$  auf den gleichen niedrigen Wert:

$$w_{ij} < \frac{L}{L-1+m}$$

$L$  ist eine Konstante  $> 1$  (meist wird  $L = 2$  gesetzt) und  $m$  ist die Anzahl der Neuronen in der Vergleichsschicht. Wählt man zu große  $w_{ij}$ , kann es passieren, dass alle Eingaben auf nur ein Neuron der Erkennungsschicht abgebildet werden.

Alle Gewichte  $w_{ij}$  der binären Matrix  $W_{ij}$  werden auf 1 gesetzt.

Der Toleranzparameter  $\rho$  wird aus dem Intervall  $[0, 1]$  festgelegt – abhängig vom gewünschten Grad der Übereinstimmung. Je größer  $\rho$  gewählt wird, desto genauer unterscheidet das Netz. Für eine perfekte Übereinstimmung wählt man  $\rho = 1$ . Meist werden Werte für  $\rho$  zwischen 0.7 und 0.99 gewählt.

### *Erkennungsphase*

Zu Beginn gilt  $I = (0, \dots, 0)$ , daher ist der Verstärkungsfaktor  $g_2 = 0$ . Das heißt alle Neuronen der Erkennungsschicht sind abgeschaltet und der Eingangsvektor  $V$  der Vergleichsschicht ist ebenfalls der Nullvektor.

Liegt ein vom Nullvektor verschiedener Eingabevektor an, erhalten die Verstärkungsfaktoren  $g_1$  und  $g_2$  den Wert 1. Jetzt können die Neuronen der Vergleichsschicht nach der 2/3-Regel feuern und der Vektor  $S$  bildet ein Duplikat der Eingabe  $I$ . Die Netzeingabe ( $t_j = \text{net}_j$ ) für jedes Neuron  $j$  der Erkennungsschicht berechnet sich aus dem Skalarprodukt seines Gewichtsvektors  $W_j$  und  $S$ .

$$u_j = \begin{cases} 1 & , \text{ falls } t_j = \langle W_j | S \rangle = \sum s_i w_{ij} = \max \\ 0 & , \text{ sonst} \end{cases}$$

Es wird nur das Neuron  $J$  mit dem größten Skalarprodukt aktiviert, alle anderen Neuronen erhalten den Wert 0. Es feuert nur das Neuron  $J$ , dessen Gewichte dem Eingabevektor am ähnlichsten sind.

### *Vergleichsphase*

Jedes Neuron  $i$  aus der Vergleichsschicht erhält ein binäres Signal  $v_i$  mit dem gleichen Wert wie  $w_{ji}$ .

$$v_i = \sum_{j \in F_2} u_j w_{ji} = w_{ji}$$

Da  $U$  vom Nullvektor verschieden ist, gilt  $g_1 = 0$ . Jetzt können nur noch Neuronen aktiviert werden, bei denen  $l_i = 1$  und  $v_i = 1$  gilt. Dadurch werden die Komponenten von  $S$  deaktiviert, deren Eingabe nicht mit dem gespeicherten Muster übereinstimmt. Dies entspricht einem Vergleich der Kategorieninformation und den dazu gehörenden Informationen des Langzeitgedächtnisses mit der tatsächlich vorliegenden Eingabe.

Die Berechnung der Ähnlichkeit der Vektoren  $I$  und  $S$  geschieht durch:

$$\frac{|S|}{|I|} = \frac{|V \wedge I|}{|I|} = \frac{|W_j \wedge I|}{|I|} \geq \rho$$

Ein Reset erfolgt, wenn diese Ungleichung nicht erfüllt ist, d.h. es gilt:

$$|S| / |I| < \rho$$

### *Suchphase*

Erfolgte in der Vergleichsphase ein Reset, wird  $g_1 = 1$  gesetzt (da  $U$  wieder gleich dem Nullvektor ist) und somit erscheint die Eingabe  $I$  wieder unverändert in  $S$ . Dadurch wird ein anderes Neuron der Erkennungsschicht aktiviert und ein anderes Muster wird an die Vergleichsschicht geliefert.

Dieser Ablauf wiederholt sich, bis eine der folgenden Abbruchbedingungen erfüllt ist:

- das gespeicherte Muster ähnelt der Eingabe ausreichend
- kein gespeichertes Muster ähnelt der Eingabe

Im ersten Fall geht das Netz in einen Trainingszyklus, durch den der zum Gewinnerneuron  $J$  der Erkennungsschicht gehörende binäre und der reellwertige Gewichtsvektor  $W_j$  modifiziert wird.

Trifft der zweite Fall ein, dann wird ein bisher nicht genutztes Neuron  $j$  aus der Erkennungsschicht ausgewählt und seine Gewichtsvektoren  $W_j$  werden an das Eingabemuster angepasst.

### *Gewichtsadaption*

Es werden zwei Arten des Trainings unterschieden:

- langsames Training (slow training, slow learning)
- schnelles Training (fast training, fast learning)

Da der Eingabevektor nur kurz anliegt, haben die Gewichte des Netzes beim langsamen Training nicht genug Zeit, ihre asymptotischen Werte zu erreichen. Die Gewichte  $w_{ij}$  besitzen hier kontinuierliche Werte und werden durch die statistische Verteilung der Eingabevektoren bestimmt. Die Netzwerkdynamik wird über Differentialgleichungen beschrieben.

Beim schnellen Training liegt der Eingabevektor so lange an, bis die Gewichte ihre stabilen Werte erreichen. Die Anpassung des reellwertigen Gewichtsvektors  $W_j$  erfolgt über die normalisierten Werte des Vektors  $S$ :

$$w_{ij} < \frac{Ls_i}{L-1 + \sum_{k \in F_1} s_k}$$

$J$  ist die Nummer des Gewinnerneurons der Erkennungsschicht,  $s_i$  ist das  $i$ -te Element des Ausgabevektors der Vergleichsschicht und  $w_{ij}$  ist das Gewicht des Vektors  $W_{*j}$ , welches das Neuron  $i$  aus  $F_1$  mit dem Gewinnerneuron  $J$  aus  $F_2$  verbindet. Die Konstante  $L$  wurde schon bei der Initialisierung erklärt.

Die Gewichtsanzpassung des binären Vektors  $W_j$  erfolgt über  $w_{ji} = s_i$ . Das Gewinnerneuron  $J$  aus der Erkennungsschicht ist mit dem Neuron  $i$  aus der Vergleichsschicht über  $w_{ji}$  verbunden. Das heißt, die Gewichte gleichen den entsprechenden Werten des binären Vektors  $S$ .

Durch die Initialisierung und den Lernalgorithmus wird sichergestellt, dass der binäre Gewichtsvektor  $(w_{j1}, w_{j2}, \dots, w_{jm})$  immer einer binären Abbildung des reellwertigen Vektors  $(w_{1j}, w_{2j}, \dots, w_{mj})$  entspricht.

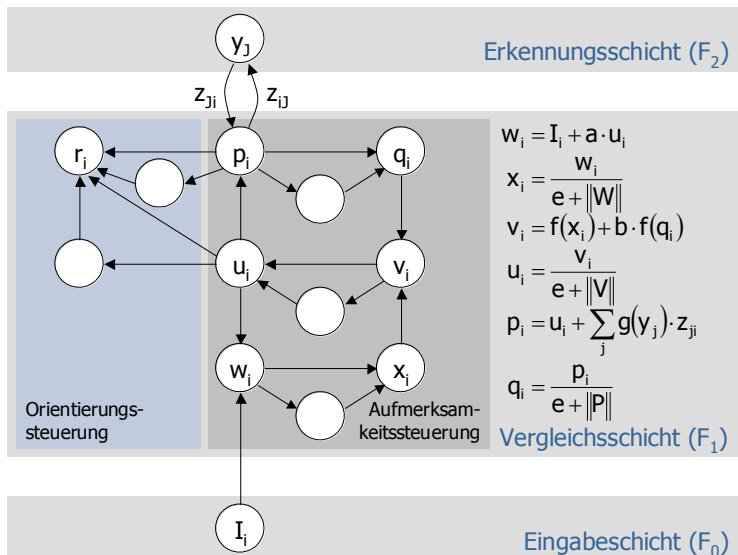
### 3. Weitere Erweiterungen der ART-Architektur

In diesem Abschnitt wird nur kurz auf weitere Architekturen eingegangen. Als vertiefende Literatur besonders zu empfehlen sind: [Zell2003], [CarGro1990] und [CaGrRo1991]. Eine umfassende Einführung zur Fuzzy-Logik und zu Fuzzy-Systemen findet man in [KrGeKl1995].

#### ART-2

ART-2 ist eine Erweiterung der ART-1-Architektur mit dem Ziel auch reellwertige Eingaben zu verarbeiten. Um solche Eingaben zu verarbeiten, wurde die Vergleichsschicht auf 3 Schichten von Neuronen mit 6 unterschiedlichen Neuronentypen erweitert.

In der nebenstehenden Abbildung ist ein Schnitt durch ein ART-2-Netz für das Gewinnerneuron  $y_j$  und die entsprechenden Berechnungen zu sehen. Die nicht näher bezeichneten Neuronen existieren nur einmal in Netz und dienen zur Berechnung der Euklidischen Norm der jeweiligen Eingabevektoren.  $Z_{ij}$  und  $Z_{ji}$  sind die Gewichtsmatrizen  $W_{ij}$  und  $W_{ji}$  aus der ART-1-Architektur.



#### ART-3

ART-3 basiert auf ART-2. Ziel dieser Architektur ist es, weitere (chemische) Eigenschaften von Nervenzellen – wie die Ansammlung von Neurotransmittern, deren Freisetzung und schließlich die

Deaktivierung der Neurotransmitter – durch Differentialgleichungen zu modellieren. Das Netz erlaubt es, Folgen asynchroner Eingabemuster zu verarbeiten. Einen umfassenden Einblick in diese Netzform bietet [CaGrRo1991].

## ART-MAP

Bei ART-MAP werden zwei ART-(1|2|3)-Netze (ART<sup>a</sup> und ART<sup>b</sup>) miteinander kombiniert, mit dem Ziel überwachtes Lernen zu realisieren. Dabei werden die Erkennungsschichten (F<sub>2</sub><sup>a</sup> und F<sub>2</sub><sup>b</sup>) beider Netze durch einen Assoziativspeicher, dem MAP-Feld F<sup>ab</sup>, verbunden. Das MAP-Feld dient zur Ausgabe der Klassifikation. Da die Lerneingabe im Netz kodiert wird, besteht die Möglichkeit, dass mehrere Trainingsmuster auf dieselbe Klasse abgebildet werden.

## Fuzzy-ART

Fuzzy-ART ist eine Erweiterung der ART-1-Architektur, die es erlaubt auch reellwertige Eingabevektoren zu verarbeiten. Es werden dabei Operationen der Fuzzy-Logik benutzt. So wird die logische Operation AND durch eine t-Norm ersetzt (in [Zell2003] wird  $T_{\min}(a,b) = \min\{a,b\}$  verwendet).

In der folgenden Tabelle werden die wichtigsten Unterschiede gegenübergestellt:

	ART-1	Fuzzy-ART
Eingabevektor	binär	analog (reellwertig)
Klassenauswahl in Erkennungsschicht	$T_j = \frac{ I \wedge W_j }{ W_j }$	$T_j = \frac{ T_{\min}(I, W_j) }{ W_j }$
Ähnlichkeitskriterium für Reset	$\frac{ I \wedge W_j }{ W_j } \geq \rho$	$\frac{ T_{\min}(I, W_j) }{ W_j } \geq \rho$
Gewichtsänderung beim schnellen Training	$W_j(t+1) = I \wedge W_j(t)$	$W_j(t+1) = T_{\min}(I, W_j(t))$

## 4. Bemerkungen

Bei allen ART-Architekturen wird ein unüberwachtes Lernen realisiert, einzig bei ART-MAP findet überwachtes Lernen statt.

Die ART-Architektur besitzt die Fähigkeit das Stabilitäts-Plastizitäts-Dilemma zu lösen:

- Das Netzwerk erlaubt eine gewisse Plastizität für neue Muster, d.h. es lassen sich neue Kategorien durch neue Muster erzeugen. Dies wird durch den zweiten Fall aus der Suchphase erreicht.
- Das Netz bietet Stabilität bereits gelernter Muster, d.h. bestehende Kategorien werden durch neue Muster nur verändert wenn sie diesen stark ähneln – sonst erfolgt keine Änderung. Dies lässt sich über den Toleranzparameter  $\rho$  und die Ähnlichkeitsbestimmung steuern.

Weitere Vorteile sind, dass eine einmalige Präsentation eines Eingabemusters ausreichend zum Lernen (fast learning) sein kann (ein Kind berührt auch nur einmal eine heiße Herdplatte!) und eine realitätsnahe Nachbildung biologischer Vorgänge (ART-3), die insbesondere von Biologen und Psychologen als Modell für Experimente (z.B. zur Untersuchung von Halluzinationen oder Autismus) verwendet wird.

Als Nachteile haben sich die folgenden zwei Fakten herausgestellt: ART ist mathematisch schwer zu beschreiben und hat eine sehr komplexe Architektur (insbesondere ART-2 und darauf aufbauende). Dies

fördert nicht unbedingt das Verständnis für die Architektur bzw. es erfordert eine hohe Einarbeitungszeit. Daher scheint ART auch nur in Nischenbereichen seine Verbreitung zu finden.

## 5. Literatur

- [CarGro1987]: G. Carpenter & S. Grossberg: A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 37, 1987  
(<http://cns-web.bu.edu/Profiles/Grossberg/CarGro1987CVGIP.pdf>)
- [CarGro1990]: G. Carpenter & S. Grossberg: ART3 – Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures, Neural Networks, Vol. 3, 1990  
(<http://cns-web.bu.edu/Profiles/Grossberg/CarGro1990NN.pdf>)
- [CaGrRo1991]: G. Carpenter, S. Grossberg, D. Rosen: Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System, Neural Networks, Vol. 4, 1991  
(<http://cns-web.bu.edu/Profiles/Grossberg/CarGroRos1991NNFuzzyART.pdf>)
- [Gross1976]: S. Grossberg: Adaptive pattern classification and universal recording – I. Parallel development and coding of neural feature detectors, Biological Cybernetics 23, 1976
- [KröSma1996]: B. Kröse & P. van der Smagt: An Introduction to Neural Networks, 8th edition, 1996  
(<http://www.robotic.dlr.de/Smagt/books/>)
- [KrGeKI1995]: R. Kruse, J. Gebhardt, F. Klawonn: Fuzzy-Systeme, Teubner, Stuttgart, 2. überarbeitete Auflage, 1995
- [Zell2003]: A. Zell: Simulation neuronaler Netze, Kapitel 22 – Adaptive Resonance Theory, Oldenbourg Wissenschaftsverlag, München, 4. unveränderter Nachdruck, 2003