

# Robotersteuerung mit RBF-Netzen

*Jörg Bruske*

Christian Albrechts Universität zu Kiel  
Lehrstuhl für Kognitive Systeme  
e-mail: jbr@informatik.uni-kiel.de

## Übersicht

Ziel unseres Vortrages zum Thema Kognitive Robotik war es, die Aufmerksamkeit der Teilnehmer auf die möglichen Vorteile des vermehrten Einsatzes von Radialen Basis Funktions (RBF) Netzen in der Robotersteuerung zu lenken.

Zu diesem Zweck begann der Vortrag mit einer kurzen Vorstellung der wesentlichen Eigenschaften von RBF Netzen in Abgrenzung zu den immer noch dominierenden mehrschichtigen künstlichen Neuronalen Netzen mit sigmoiden Aktivierungsfunktionen. Die Nähe zur Fuzzy Logic und zur Bayesschen Funktionsapproximation sowie ihre Ableitung aus der Regularisierungstheorie tragen wesentlich zum besseren Verständnis der Funktionsweise von RBF Netzwerken bei und ermöglichen die Entwicklung effizienter Trainingsprozeduren. RBF Netze erlauben in einfacher Weise ihre Vorstrukturierung mit a priori Wissen, auf der anderen Seite können nach Abschluß des Trainingsvorganges die funktionalen Zusammenhänge genau so einfach in Form von Regeln extrahiert werden.

Obwohl zur Zeit der Schwerpunkt des Einsatzes von RBF Netzen (wie auch von MLPs) mehr in der Klassifikation und Mustererkennung denn in der Steuerung von Robotern liegt, gibt es auch dort bereits mehrere interessante Anwendungen. Von diesen wurden exemplarisch vier andiskutiert.

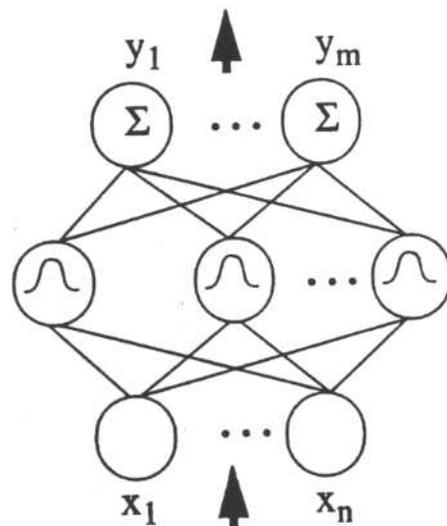
Als eine Erweiterung des reinen Radialen Basis Funktionsansatzes wurden die an unserem Lehrstuhl entwickelten Dynamischen Zell Strukturen (DCS) vorgestellt. Diese erlernen gleichzeitig mit der Approximation einer Funktion die Topologie der Eingabemannigfaltigkeit und nutzen diese ähnlich einer Kohonenkarte. Die Vielzahl verwandter Algorithmen erfordert eine Abgrenzung des DCS Ansatzes.

Den Schluß des Vortrages bildete ein Ausblick auf den aktuellen Forschungsgegenstand des Autors, nämlich den Einsatz von RBF-Netzen zum Lernen in verhaltensbasierten Roboterarchitekturen.

Dieser Beitrag ergänzt die Vortragsvorlagen durch Text und Literaturhinweise.

## Radiale Basis Funktions Netzwerke

RBF Netzwerke haben eine dreischichtige Struktur: Eine Eingabeschicht, eine versteckte Schicht von Knoten mit einer lokalen, radial streng monoton fallender Aktivierungsscharakteristik und eine Ausgabeschicht meist linearer Knoten.



z.B.  $\text{exp}\left(-\frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i)\right)$

Die Ausgabe eines RBF Netzes berechnet sich als mit der jeweiligen Aktivierung gewichtete Summe der mit den Knoten der versteckten Schicht assoziierten Vektoren  $o_i$ :

$$y = \sum_i o_i rbf_i(x)$$

Oft erfolgt die Ausgabeberechnung auch mit dem sogenannten "Partitioning to One" bzw. "Normalisierten RBFs" gemäß

$$y = \frac{\sum_i o_i rbf_i(x)}{\sum_i rbf_i(x)} = \sum_i o_i \frac{rbf_i(x)}{\sum_j rbf_j(x)}$$

Letzteres Berechnungsschema bietet oft günstiger Interpolationseigenschaften, da es das "Durchsacken" zwischen Stützstellen verhindert.

Vergleichend mit MLPs läßt sich feststellen:

- MLPs und RBFs sind "universale Approximatoren", d.h. sie können jede stetige Funktion beliebig genau approximieren.
- Parameteränderungen in RBFs wirken *lokal* (in MLPs: *global*). Dies hat z.B. die Konsequenz, daß in RBF Netzen neue versteckte Knoten eingefügt werden können, ohne daß das Netz komplett neu trainiert werden muß (inkrementelles Training).
- RBFs als erweitertes "Table Lookup": Die mit den versteckten Knoten verknüpften Ausgabevektoren  $o_i$  lassen sich als Tabelleneintrag zum korrespondierenden Aktivierungszentrum interpretieren. Zwischen Tabelleneinträgen wird interpoliert.
- Für RBFs existieren sehr schnelle Hybride Trainingsverfahren.

## RBF Besonderheiten

### Nähe zur Fuzzy-Logic:

Es läßt sich relativ einfach zeigen<sup>1</sup>, daß eine ganze Klasse von Fuzzy-Reglern funktional äquivalent mit normalisierten RBF-Netzen ist. Hierbei realisiert jede Radiale Basis Funktion  $rbf_i$  die Prämisse einer Fuzzy-Regel, und jeder Ausgabevektor  $o_i$  repräsentiert die Konsequenz einer Fuzzy-Regel. Dies ermöglicht einerseits die Vorstrukturierung/Initialisierung eines RBF-Netzes mit Vorwissen in Form von Fuzzy-Regeln, andererseits kann ein RBF-Netzwerk im Sinne eines Fuzzy-Entscheidungssystem interpretiert werden.

Desweiteren impliziert diese funktionale Äquivalenz, das Fortschritte auf dem einen Gebiet (z.B. neue adaptive Verfahren, neue Resultate bzgl. der Ausdruckskraft) sich oft unverändert auf das andere Übertragen lassen.

### Nähe zu Approximation mit Gaußscher Mischverteilung:

Die im Sinne der Bayesschen Approximationstheorie beste Approximation einer Funktion ist ihr Erwartungswert. Unter bestimmten Annahmen<sup>2</sup> läßt sich dieser schreiben als

$$y(x) = E(y|x) = \dots = \frac{\sum_i y_i P(x|\mu_i)}{\sum_i P(x|\mu_i)}$$

Falls sich die bedingten Wahrscheinlichkeitsverteilungen nun als Gaußverteilungen modellieren lassen, so läßt sich diese Gleichung wiederum als normalisiertes RBF-Netzwerk interpretieren. Umgekehrt lassen sich die Parameter der Radialen Basis Funktionen oft als Parameter bedingter Wahrscheinlichkeitsverteilungen deuten. Insbesondere können Parameterschätzverfahren für Mischverteilungen zur Initialisierung von RBF-Netzen dienen.

### RBF-Netzwerke als Regularisierungs-Netzwerke

Die Aufgabe von Regularisierungsansätzen ist es, gegeben ein Menge von N Datenpunkten  $\{(x_i, y_i)\}$  und einen differentiellen Operators  $P$ , diejenige Funktion  $f$  zu finden, die das Funktional

$$H[f] = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \|Pf\|^2$$

minimiert, d.h. die Daten möglichst glatt interpoliert. Poggio und Girosi zeigen nun in ihrem vielzitierten Artikel<sup>3</sup>, daß sich als allgemeine Lösung dieser Gleichung RBF-Netze mit radialsymmetrischen Aktivierungsfunktionen ergeben, wenn  $P$  translations- und rotationsinvariant ist. .

- 
1. J. Jang, Functional equivalence between radial basis function networks and fuzzy inference systems, ohio neuroprose archive.
  2. V. Tresp, Neural Control for Rolling Mills: Incorporating Domain Theories to Overcome Data Deficiency, NIPS 4, 659-667, 1991.
  3. T. Poggio u. F. Girosi: Networks for Approximation and Learning, Proc IEEE Vol. 78, No. 9, 1481-1497, 1990

## **Anwendungsbeispiele**

In diesem Abschnitt werden einige aus Sicht des Autors vielversprechende Anwendungen von RBF-Netzwerken in der Robotik vorgestellt:

### **RBF-Netzwerk zur Walzkraftsteuerung (Siemens AG)**

Charakteristisch für diese Anwendung ist, daß Training nur sehr spärlich vorliegen, voraussetzt sind und ihre Erhebung mit hohen Kosten verbunden ist. Die im vorhergehenden Abschnitt angesprochene Nähe von RBF-Netzwerken zu Bayesschen Approximatoren wird hier genutzt, um die Parameter des Netzwerkes unter Ausnutzung der wenigen Daten off-line zu generieren.

Artikel: V. Tresp, Neural Control for Rolling Mills: Incorporating Domain Theories to Overcome Data Deficiency, NIPS 4, 659-667, 1991.

### **RBF-Netzwerke zum Erlernen der inversen Kinematik**

Die klassische Aufgabe für ein Neuronales Netz, die inverse Kinematik eines Roboterarms zu lernen, wird hier für einen Roboterarm mit sechs Freiheitsgraden gelöst. Zum Einsatz kommt ein RBF-Netzwerk, daß während des Trainings inkrementell wächst. Das Einfügen neuer Knoten erfolgt Fehler-getrieben.

Artikel: J. Hakala, R.Eckmiller, "Node Allocation and Topographical Encoding (NATE) for Inverse Kinematics of a Redundant Robot Arm for Inverse Kinematics of a 6-DOF Robot Arm", ICANN93, 309-312, 1993.

### **RBF-Netzwerke zur Trajektoriengenerierung**

Ein RBF-Netzwerke erlernt hier off-line ein Vorwärts-Modell, während die inverse Sensor-Motor Transformation on-line durch Gradientenabstieg auf einer das Vorwärtsmodell nutzenden Potentialfunktion realisiert wird.

Artikel: P.Morasso, V.Sanguineti, T. Tsuji, "Neural Architecture for Robot Planning", ICANN93, 256-261, 1993.

### **RBF-Netzwerke zur Stablen Adaptiven Regelung**

Eine stabile adaptive Regelung eines Roboterarms wird hier anhand der Aufgabe demonstriert, vorgegebenen Gelenkwinkel-Trajektorien zu folgen. Stabilität wird durch eine Feedbackschleife erreicht, in die das RBF-Netzwerk als Modell-basierter Regler integriert ist.

Artikel: J. Beerhold, M. Jansen, R. Eckmiller, "First Results on Stable Adaptive Robot Control with RBF Networks", ICANN93, 297-301, 1993.

## Dynamische Zell Strukturen

Die an unserem Lehrstuhl entwickelte Familie der Dynamischen Zell Strukturen (DCS) bezeichnet RBF-Netzwerke, die gleichzeitig mit der Adaption ihrer Parameter ein potentiell *perfekt topologierhaltendes* laterales Verbindungsnetzwerk zwischen den RBF-Knoten aufbauen und dieses wiederum zur Steuerung der Adaption und Approximation nutzen, [Bruske95].

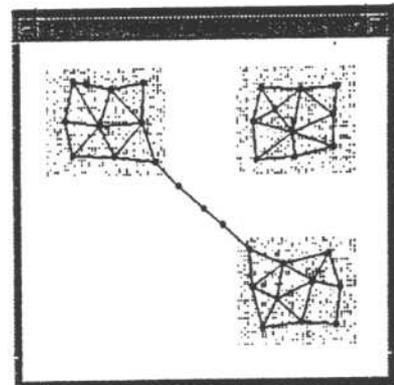
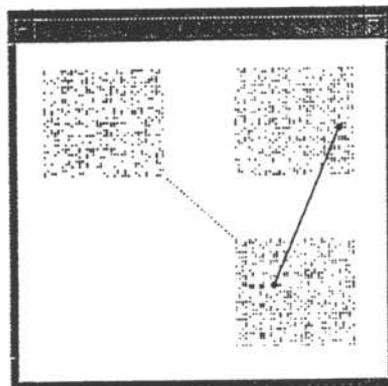
Für Neuronale Netzwerke, die ein *perfekt topologierhaltendes* laterales Verbindungsnetzwerk aufbauen, wurde von T. Martinetz der Name *Topology Representing Networks* eingeführt, [Martinetz94].

Der DCS-GCS Algorithmus, der im wesentlichen den Growing Cell Structure (GCS) Algorithmus von B. Fritzke implementiert, [Fritzke95], jedoch dessen laterales Verbindungsnetzwerk aus k-dimensionalen Simplices durch Perfekt-Topologie-Lernen ersetzt, erwies sich seinem Vater GCS auf einer Reihe von Benchmarks als ebenbürtig und konventionellen Modellen überlegen, [Bruske95]. Seine Lernprinzipien sind

- *Hebb'sches Wettbewerbslernen* zum Aufbau des lateralen Verbindungsnetzwerkes (siehe [Martinetz94])
- Aktivierung und Adaption nur von Knoten, die bzgl. des lateralen Verbindungsnetzwerkes benachbart sind
- Einfügen neuer RBF-Knoten gesteuert durch *lokalen Fehler* und das laterale Verbindungsnetzwerk (siehe [Fritzke95])

### Beispiel für unüberwachtes DCS-GCS Lernen:

Die Eingabemannigfaltigkeit besteht in diesem Beispiel aus drei Quadraten (zweidimensional), und einer eindimensionalen Verbindungslinie zwischen zwei der Quadrate. Der DCS-GCS Algorithmus, hier zur Vektorquantisierung eingesetzt, startet mit 2 verbundenen Knoten (linkes Bild). Nach Einfügung von weiteren 27 Knoten hat der DCS-GCS Algorithmus ein perfekt topologieerhaltendes Verbindungsnetzwerk zwischen den Knoten aufgebaut (rechtes Bild).



## Abgrenzung

Die Vielzahl der auf Topologierhaltung und/oder Radialen Basis Funktionen aufsetzenden Ansätze erfordert eine Abgrenzung:

### zu Kohonen's Selbstorganisierenden Merkmalskarten (SOM):

- SOM [Kohonen87] ist im allgemeinen *nicht* (perfekt) topologieerhaltend
- SOM bewährt in der *Visualisierung* von hochdimensionalen Daten

### zu Fritzke's Wachsenden Zellstrukturen (GCS):

- GCS benutzt *vordefinierte laterale Verbindungsstruktur* (k-dimensionale Simplices)
- GCS ggf. vorteilhaft, falls  $\dim(M)$  bekannt und konstant über M ist.

### zu Martinetz' Neural Gas + Hebb'scher Regel (NGH):

- NGH [Martinetz94] nutzt die laterale Verbindungsstruktur *nicht* zur Bestimmung der k topologisch nächsten Nachbarn. Adaption der Zentren und Interpolation erfolgt nur durch Neural Gas. Die Anwendung liegt in der "Pfad erhaltenden Repräsentation von Mannigfaltigkeiten".

### zu herkömmlichen RBF-Netzwerken

- DCS lernt und nutzt zusätzlich topologische Information in Form einer Perfekt Topologieerhaltenden Karte (Aufbau dieser Karte: "gratis" bzgl. Zeitaufwand und quadratisch in der Anzahl der Knoten bzgl. Speicherplatz).
- verspricht vorteilhafte Interpolationseigenschaften hinsichtlich:
  - Effizienz (Aufsuchen der nächsten Nachbarn)
  - Genauigkeit (nur über benachbarte Knoten)
- Laterales Verbindungsnetzwerk ermöglicht Einsicht in die Struktur der Eingabemannigfaltigkeit.

## Referenzen

- [Bruske95] Joerg Bruske und Gerald Sommer, *Dynamic Cell Structure learns Perfectly Topology Preserving Map*, Neural Computation, Vol.7, No.4, 1995.
- [Fritzke95] B. Fritzke, *Growing Cell Structures - a self organizing network for unsupervised and supervised training*, Neural Networks, Vol.7, No.9, 1441-1460, 1995.
- [Kohonen87] Teuvo Kohonen, *Adaptive, associative, and self-organizing functions in neural computing*, Applied Optics, Vol.26, pp. 4910-4918, 1987.
- [Martinetz94] Thomas Martinetz and Klaus Schulten, *Topology Representing Networks*, Neural Networks, No. 7, Vol. 3 , pp. 505-522, 1994.
- [Moody89] J.Moody, C.J. Darken, *Fast Learning in Networks of Locally-Tuned Processing Units*, Neural Computation, Vol. 1, No. 2, Summer 1989.



Die sensorische Information  $y(t)$  wird zunächst einer Vorverarbeitungsstufe zugeführt (diese kann ein Gedächtnis enthalten) und zu den für das Modul relevanten Features  $i(t)$  verdichtet. Übersteigt die Reinforcement-Funktion  $R(i(t))$  des Moduls nun einen Schwellwert (Schmerz), so wird das Modul aktiviert und der Regler des Moduls bestimmt mit  $u^0(i(t))$  die nächste Aktion. Höherliegende Module mit geringerer Priorität werden inhibiert. Liegt  $R(i(t))$  hingegen unter dem Schwellwert, so wird das betreffende Modul überbrückt: Die Kontrolle wird an die Module niedriger Priorität weitergegeben und deren Aktion  $u^{+1}(t)$  durchgeschaltet. Zur Realisierung der Regler werden mit Fuzzy-Logic vorstrukturierte RBF bzw. DCS Netze eingesetzt.

Adaptivität der Module wird durch die a priori vorgegebene, differenzierbare Reinforcement-Funktion  $R(\cdot)$  und ein zu erlernendes Modell der Wechselwirkung des Moduls mit der Umwelt erreicht. Letzteres ermöglicht es, aus dem Gradienten des Reinforcements nach den sensorischen Größen (die *Intention*) den Gradient des Reinforcements nach den Aktionen zu berechnen, und kann überwacht gelernt werden. Dieser Gradient wiederum läßt sich direkt zur überwachten Adaption des Reglers nutzen. Unser Ansatz ähnelt damit dem Distal Supervised Learning [Jordan92].

Vorwissen fließt in die Modularisierung (+ Hierarchisierung) von Verhalten ein, ebenso in die Sensorvorverarbeitung und die Wahl geeigneter Reinforcement-Funktionen. Vorwissen kann (muß aber nicht) zur Vorstrukturierung der RBF-basierten Reglers und der partiellen Weltmodelle eingebracht werden.

Zur Zeit testen wir die Entwicklung adaptiver Verhaltensmodule anhand von Ultraschall basierten Hindernisvermeidungs-, Wandfolge- und Zielverfolgungsaufgaben.

## Referenzen

- [Aloimonos88], Y. Aloimonos and I. Weiss and A. Bandopahay, *Active Vision*, Int'l J. Comp. Vision, No. 7, 333-356, 1988.
- [Brooks86], R. A. Brooks, *A Robust Layered Control System For A Mobile Robot*, IEEE J. of Robotics and Automation, Vol. 2, No. 1, 14-23, 1986.
- [Jordan]
- [Mataric92], M. J. Mataric, *Integration of Representation Into Goal-Driven Behavior-Based Robots*, IEEE Trans. on Robotics and Automation, Vol.8, No.3, 304-312, 1992.
- [Mataric93], R. A. Brooks and M. J. Mataric, *Real Robots, Real Learning Problems*, in Robot Learning, 193-214, Kluwer Academic, 1993.
- [Jordan92], M.I. Jordan and D.E. Rumelhart, *Forward Models: Supervised Learning with a Distal Teacher*, Cognitive Science, Occasional Paper #40, 1992.