

Soft-Computing II

Zusammenfassung

Aless Lasaruk

7. November 2005

Zusammenfassung

Dieses Dokument enthält Fragen und Antworten zum Fach Soft-Computing II. Die Fragen sind durch systematisches Durchgehen des gleichnamigen Skriptums entstanden, decken aber den Stoff nicht vollständig. Man beachte auch, dass die Antworten nicht exakt formuliert sind. Das Ziel der Formulierung ist das Minimieren der schriftlichen Antwort und Maximieren der mündlichen.

Inhaltsverzeichnis

1 Radiale-Basisfunktionen-Netze	1
1.1 Aufbau der RBF-Netze	2
1.2 Training von RBF-Netzen	2
2 Anwendungen von RBF-Netzen	3
2.1 Hinweise zur Anwendung von RBF	3
2.2 Vergleich zwischen RBF und MLP	3
2.3 Technische Anwendungn von RBF	4
3 Dynamische neuronale Netze	4
3.1 Architektur von dynamischen neuronalen Netzen	5
3.2 Training von DYNN	6
3.3 Abschliessende Bemerkungen	7
4 Anwendungen von dynamischen neuronalen Netzen	7

1 Radiale-Basisfunktionen-Netze

- **Zu welcher Klasse von Netzen gehören *RBF*-Netze?**
Überwacht trainierte, statische Netze.
- **Was ist der Unterschied zwischen MLP und RBF?**
RBF haben immer drei Schichten in denen statt Linearkombination (Skalarprodukt) der euklidische Abstand berechnet wird. MLP benutzen sigmoide Aktivierungsfunktionen. RBF Radialsymmetrische.
- **Welche Vorteile haben RBF-Netze?**
RBF-Netze besitzen außerhalb von Bereichen, wo sie trainiert wurden keine unkontrollierbaren Ausgaben. Direkte Berechnung der Gewichte zwischen der versteckten und der Ausgabeschicht durch ein lineares Ausgleichsproblem.
- **Wie kann man RBF-Netze anwenden?**
Modellierung nichtlinearer kontinuierlicher Funktionen, Klassifikation.

1.1 Aufbau der RBF-Netze

- **Welche Propagierungsfunktionen besitzen die Neuronen der Ausgabeschicht?**
Linearkombination aus Gewichten und der Ausgabe der Eingabe und der versteckten Schicht.
- **Welche Propagierungsfunktionen besitzen die Neuronen der versteckten Schicht?**
Euklidischer Abstand zwischen Eingabe und dem Gewichtsvektor.
- **Welche Aktivierungsfunktion wird den Neuronen der Ausgabeschicht zugeordnet?**
Sigmoidale oder identische Abbildung.
- **Welche Aktivierungsfunktion wird den Neuronen der versteckten Schicht zugeordnet?**
Eine *radialsymmetrische Basisfunktion*.
- **Sind bei RBF schichtübergreifende Verbindungen erlaubt?**
Ja.
- **Welche Beispiele für Basisfunktionen von RBF kennen Sie?**
Gauss-Funktion (e^{-x^2/p^2}), Multiquadratische Funktion ($\sqrt{(x^2 + p^2)}$), Inverse multiquadratische Funktion ($\frac{1}{\sqrt{x^2 + p^2}}$).
- **Ist die Funktion des Netzes linear?**
Nein. Nicht einmal, wenn die Aktivierungsfunktion der Ausgabeschicht und versteckten Schicht linear ist.
- **Sind die Basisfunktionen immer gleich?**
Nein. Im Allgemeinen kann man sinnvoll mit dem Parameter der Funktion die Basisfunktionen nachtunen.
- **Ist die Skalierung der Eingabe/Ausgabedaten erforderlich?**
Nein. Wie bei SOM.

1.2 Training von RBF-Netzen

- **Wie Trainiert man RBF-Netze?**
Man bestimmt zunächst (unüberwacht) die Zentren der Basisfunktionen. Dann bestimmt man die Gewichte zwischen der versteckten und der Ausgabeschicht. Dann bessert man das Ergebnis mit einem Gradientenabstieg nach.
- **Warum ist der Schritt 3 nötig?**
Die Werte der schichtübergreifenden Gewichte und Schwellwerte sind nicht fest. Andere entstammen vereinfachten Annahmen.
- **Wie initialisiert man die Gewichte schichtübergreifender Verbindungen?**
Mit zufälligen kleinen Werten.
- **Wie werden die Parameter im Schritt 1 eingestellt?**
Man ordnet jedem Gewicht paarweise verschiedene Eingabemuster zu. Man bestimmt dann den maximalen Abstand d zweier Gewichtsvektoren. Dann ordnet man jedem Parameter den Wert $\frac{d}{\sqrt{U}}$ mit U die Anzahl der versteckten Neuronen.
- **Was ist das Ergebnis des Trainingsschrittes?**
Für jede Eingabe kann die Aktivierung jedes versteckten Neurons als Funktionswert einer Basisfunktion berechnet werden.
- **Welche anderen Möglichkeiten der Initialisierung bestehen?**
Mit ersten Mustern (ohne Schritt 3 schlechte Ergebnisse), Vorverarbeitung mit lernender *Vektorquantisierung*, *Clustering Verfahren*, SOM, überwachte Lernverfahren.
- **Warum wird ein einfaches Verfahren verwendet?**
Nachtrainieren im Schritt 3, Lösung des Ausgleichsproblems muß existieren.

- **Was passiert im Schritt 2, wenn das Netz mehrere Ausgabeneuronen hat?**
Für jedes muß ein *Ausgleichsproblem* gelöst werden.
- **Warum muß ein Ausgleichsproblem durchgeführt werden?**
Die Anzahl der (approximations)Stellen (also Lernmuster) ist größer als die Anzahl der einstellbaren Parameter.
- **Können Sie das Ausgleichsproblem formulieren?**
 $Aw = b$, wobei A die Matrix, die aus den Werten der Basisfunktionen für die Muster besteht, w der Gewichtsvektor, der zu bestimmen ist und b sind die Wunschausgaben.
- **Wie löst man das Ausgleichsproblem?**
Zum Beispiel mit Hilfe der *Normalengleichung*. Es gibt aber errizientere numerisch stabilere Methoden.
- **Wie kann man Glattheit der Ausgabefunktion einstellen?**
Man addiert einen Strafter an die Wertefunktion, der hohe Änderungen bestraft. Oder durch ausreichende Anzahl an Eingabemustern.
- **Mit welchen Werten werden die Gewichte im Schritt 3 initialisiert?**
Alle, die bereits bestimmt wurden mit den bestimmten und die restlichen mit kleinen zufälligen.
- **Mit welcher Methode wird im Schritt 3 trainiert?**
Backpropagation, wie MLP. Das geht wegen der stetigen Differenzierbarkeit der Netzfunktion.
- **Was sollte man beachten im Schritt 3, damit man auch andere Aktivierungsfunktionen verwenden kann?**
Die sollten der identischen Abbildung ähnlich sein.
- **Um welche geometrischen Figuren handelt es sich bei RBF-Netzen?**
Vereinigungen von *Hyperelipsoiden* bzw. *Hypersphären*.

2 Anwendungen von RBF-Netzen

- **Was ist der Unterschied im Approximationsverhalten von RBF und MLP?**
Bei RBF tragen die Werte der Aktivierungsfunktionen außerhalb bestimmter Bereiche nicht zum Ergebnis bei. Als approximiert ein RBF Netz lokal. Im Gegensatz dazu approximiert (versucht) MLP global.

2.1 Hinweise zur Anwendung von RBF

- **Welche Punkte sind bei der Anwendung wichtig?**
Selektion, Skalierung und Kodierung von Eingabegrößen, Wahl der Parameter der Architektur, Anwendung mit dem Lernziel der Generalisierung, Kriterien zur Bewertung trainierter RBF-Netze.
- **Wann ist die Skalierung der Ausgaben bei RBF nötig?**
Wenn man sigmoide Aktivierungsfunktionen in der Ausgabeschicht verwendet.
- **Kann ein RBF-Netz auch overfitten?**
Ja. Genau so, wie andere Netze.
- **Generalisiert ein RBF-Netz in jedem Fall?**
Nein. Außerhalb der Trainierten Bereiche sehr unwahrscheinlich.

2.2 Vergleich zwischen RBF und MLP

- **Was sind nennenswerte Unterschiede?**
Aktivierungsfunktionen sind radialsymmetrisch, Euklidischer Abstand statt Skalarprodukt, Hypersphären statt Hyperebenen, Trainieren aller Parameter gemeinsam bei MLP, Unterschiedliche Neuronentypen bei RBF.

- **Wie viele quantitativ Neuronen tragen zur Ausgabe bei bei RBF?**
Sehr wenige.
- **Wie wird die XOR Funktion bei MLP und RBF realisiert?**
Bei RBF wird im Raum der Aktivierungen der Neuronen linear separiert. Die Aktivierungen stellen Sphären um die Wahr-Werte. Bei MLP sind zwei Flächen, die auseinanderlaufen, parallel zur Richtung der Falsch-Werte.
- **Welche Beziehung besteht zwischen Spärlich separierbar und linear separierbar?**
Aus Sphärisch separierbar folgt linear separierbar. Die Umkehrung folgt allerdings nicht.
- **Kann man die XOR mit einem Neuron und RBF darstellen?**
Mit allgemeinen Gauss-Funktionen.

2.3 Technische Anwendungen von RBF

- **Welche typische Anwendungen von RBF-Netzen kennen Sie?**
Zeitreihenanalyse, Bildverarbeitung, Sprachverarbeitung, Radarüberwachung.
- **Wie kann man mit RBF Farbkalibrierung durchführen?**
Man bestimmt für Testwerte der Farben Abweichungen von der *Kennlinie* (Kennlinien-linearisierung).
- **Wie entsteht eine Kennlinie?**
Sollausgabe/tatsächliche Ausgabe.
- **Sind RBF-Netze den MLP's vorzuziehen?**
Nein. Es hängt von der Anwendung ab.

3 Dynamische neuronale Netze

- **Was ist der Unterschied zwischen *Speicherfähigkeit* und *Rekursion*?**
Bei der Speicherung hält das System Eingaben und interne Zustände des Netzes für beliebige Zeit, welche zu jeder Zeit benutzt werden (hier Verzögerung in Systemen ohne Rückkopplungen). Rekursion sind Rückkopplungen (hier immer mit Speichern verbunden).
- **Welches Modell ist im Kontext der Vorlesung stärker: Rekursion oder Speicherung?**
Rekursion, weil diese Auswirkungen auf beliebig lange zukünftige Zustände des Netzes hat.
- **Gibt es biologische Vorbilder für dynamische neuronale Netze?**
Signale über Axonverbindungen weisen oft unterschiedliche Zeitabstände der Ankünfte.
- **Was ist der zentrale Unterschied zwischen bisherigen Netzen und DYNN?**
Die bisherigen waren statisch.
- **Was sind TDNN-Netze?**
Erweiterung von MLP durch Einführen von *FIR-Synapsen*.
- **Was sind NARX-Netze?**
Erweiterung von MLP durch Einführen von Rekursionsverbindungen und Verzögerungen von Rückkopplungssignalen und externen Eingaben.
- **Wie kann man dynamische Netze verwenden?**
Zeitreihenvorhersage, Klassifikation von Zeitreihen, Abbildung einer Zeitreihe auf eine andere, Regler, Modellierung von Zeitreihen.

3.1 Architektur von dynamischen neuronalen Netzen

- **Was unterscheidet die formale Struktur eines *DYNN* von einem *MLP*?**
Es gibt eine rekurrente Schicht mit Verzögerungen. Gewichte werden zu Gewichtsvektoren, welche Verzögerung symbolisieren.
- **Welche Aktivierungsfunktionen verwendet man bei *DYNN*?**
Sigmoidale stetig differenzierbare streng monoton steigend.
- **Mit welchen Gewichten sind rekurrente Verbindungen versehen?**
1.
- **Wie kann man *DTNN* als Spezialfall von *DYNN* definieren?**
Bei *DTNN* entfallen rekurrente Neuronen samt der Verbindungen.
- **Wie kann man *NARX* als Spezialfall von *DYNN* definieren?**
Es gibt keine Verzögerungen bei allen FIR-Filtern, außer an der Eingabeschicht und an der rekurrenten Schicht.
- **Was ist ein Haupteinsatzbereich von *TDNN*?**
Translationsinvariante Erkennung von Mustersequenzen.
- **Was ist ein *MLP* mit einem gleitenden Zeitfenster?**
Ein *DYNN* ohne rekurrenter Schicht und mit Verzögerungen zwischen der Eingabeschicht und der ersten verdeckten Schicht.
- **Wie kann man übliche Backpropagation für *MLP-sw* verwenden?**
Indem man einfach das Eingabefenster zeitlich entfaltet.
- **Was ist ein *rezeptives Fenster*?**
Zeit, in der dem Netz Informationen zur Verfügung stehen.
- **Wie groß sind rezeptive Fenster bei *DYNN*'s?**
Potenziell unendlich lang.
- **Gibt es zu jedem *TDNN* ein äquivalentes *MLP-sw*?**
Ja. Man kann das *TDNN* zeitlich entfalten. Man beginnt z.B. bei der Ausgabeschicht und ersetzt nach unten zur Eingabeschicht alle FIR Verbindungen durch neue Neuronen.
- **Welche Probleme bekommt man bei der Anwendung der zeitlichen Entfaltung?**
Sehr viele Gewichte und gemeinsam genutzte Gewichte. Mehr Freiheitsgrade.
- **Kann man auch umgekehrt von der Eingabe- zur Ausgabeschicht bei der Entfaltung vorgehen?**
Ja.
- **Wann sind ein *TDNN* und *MLP-sw* bei gleichem rezeptiven Fenster funktional äquivalent?**
Wenn sie auch strukturell äquivalent sind.
- **Welche Folgen hat die Vergrößerung der Zahl der Freiheitsgrade?**
Gefahr der Überanpassung, längere Trainingszeiten, mehr Trainingsmuster.
- **Was bedeutet eigentlich *NARX*?**
Nonlinear autoregressive model (Rückkopplungen) with exogenous inputs (gleitendes Fenster).
- **Für welche Anwendungen sind *NARX*-Netze gut geeignet?**
Analyse von Langzeitauswirkungen.
- **Zu welchen Berechnungsmodellen sind *NARX*-Netze äquivalent?**
Turing Maschinen.

3.2 Training von DYNN

- **Was unterscheidet feste Lernaufgaben bei MLP und bei DYNN?**
Bei DYNN ist die Reihenfolge der Muster von der Aufgabe abhängig, bei MLP von einer Lernstrategie.
- **Was versteht man unter *inkrementellen Lernen*?**
Lernen einer festen Lernaufgabe aus mehreren Sequenzen.
- **Mit welchen Einschränkungen kann man ein DYNN musterweise trainieren?**
Sehr geringe Lernrate.
- **Wie heißt der Lernalgorithmus für DYNN?**
Temporal Backpropagation.
- **Was unterscheidet den Lernalgorithmus bei DYNN und bei MLP?**
Man muss zunächst die ganze Eingabesequenz durch das Netz propagieren. Dann wird die Gewichtsänderung anfangend mit dem letzten Muster durchgeführt.
- **Welche Daten muss man zwischenspeichern?**
Aktivierungen der Neuronen für alle Zwischenschritte.
- **Wie kann man Initialisierungsprobleme lösen?**
Indem man einen Teil der Eingabesequenz wiederholt.
- **Wieso ist beim Trainieren entfalteter TDNN das *Lokalitätsprinzip* verletzt?**
Gemeinsame Betrachtung von Gewichten erfordert eine globale Sicht.
- **Welches Fehlermaß wird bei der Netzbewertung verwendet?**
Gesamte quadratische Fehler über alle Muster.
- **Welche Näherung wird dabei betrachtet?**
Konstanter Gewichtsvektor (für epochenweises Lernen passend).
- **Was ist das Fehlersignal hier?**
Ableitung des Fehlers nach der Eingabe der Schicht.
- **Wie wird die Suchrichtung bei Gewichten berechnet?**
Negative des Gradienten des Fehlers nach dem Gewicht für ein Muster: Ableitung des Fehlers nach Gewicht zwischen i und j über die Netzeingabe von j .
- **Über welchen Zwischenschritt wird mit der Kettenregel das Fehlersignal in der äußeren Schicht bestimmt?**
Über die Aktivierung der Schicht.
- **Über welchen Zwischenschritt wird mit der Kettenregel das Fehlersignal für verdeckte Schichten bestimmt?**
Über Eingaben der nachfolgenden Schichten für zukünftige Muster und anschließend über die Aktivierung der Schicht.
- **Wie verändert sich das Fehlersignal?**
Der wird zum Fehlervektor, der auch zukünftige Fehlersignale enthält.
- **Wie kann man den Namen temporal Backpropagation erklären?**
Die Fehlersignale werden rückwärts über die FIRS ähnlich, wie bei MLP propagiert.
- **Wie verhält man sich bei nicht verfügbaren Fehlersignalen?**
Man nimmt an, sie wären 0.
- **Kann man auch normale Backpropagation auf TDNN anwenden?**
Ja, aber die Lernrate muss sehr klein gewählt sein, damit tatsächliche Gradient angenähert wird.
- **Auf welcher Idee basiert der Trainingsalgorithmus für DYNN?**
Zeitliche Netzentfaltung und gemeinsame Gewichtsnutzung.
- **Wie lange wird das Netz zeitlich entfaltet?**
Bis zur Anzahl der Muster.

- **Was ist der Unterschied zum Trainingsalgorithmus für MLP?**
Betrachtung der Fehlersignale in Unterschiedlichen Netzen und Fehlerinjektion durch Addieren des Fehlersignals in die Oberschicht der inneren Netze.
- **Welche zu MLP analoge Probleme treten beim Training von DYNN auf?**
Schwere wahl der Parameter, Startpunktwahl, Plateaus, langsame Adaptation der Gewichte nahe der Eingabeschicht, lokale Minima.

3.3 Abschliessende Bemerkungen

- **Wann sollte man rekurrente und wann verzögernde Netze einsetzen?**
Rekurrente, wenn es darum geht die gesamte Entwicklung einer Reihe zu erfassen. Verzögernde bei translationsinvarianten Mustererkennung.
- **Welche weitere verzögernde Netztypen kennen Sie?**
Kontinuierliche Netze ATNN.

4 Anwendungen von dynamischen neuronalen Netzen

-
-
-
-

Index

Ausgleichsproblem, 3
Clustering Verfahren, 2
DYNN, 5
FIR-Synapsen, 4
Hyperelipsoiden, 3
Hyperspären, 3
inkrementellen Lernen, 6
Kemlinie, 4
Lokalitätsprinzip, 6
MLP, 5
NARX, 4, 5
Normalengleichung, 3
radialsymmetrische Basisfunktion, 2
RBF, 1
Rekursion, 4
rezeptives Fenster, 5
Speicherfähigkeit, 4
TDNN, 4
Vektorquantisierung, 2