

Moderne Methoden der KI: Maschinelles Lernen

Prof. Dr. sc. Hans-Dieter Burkhard
Vorlesung Sommer-Semester 2007

Neuronale Netze
(Ergänzungen zur Grundvorlesung)

Einführung/Grundlagen

Neuronale Netze sind biologisch motiviert
Neuronale können diskrete, reell-wertige und Vektorwertige Funktionen berechnen
Bevorzugt anwendbar für komplexe Daten
Meist mehrschichtige Architektur notwendig
Lernverfahren auf der Basis von Fehleroptimierung (Gradientenabstieg), z.B. Back-Propagation

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

2

Modell

Neuron:

Mehrere Eingänge, 1 Ausgang
Outputfunktion

- Eingang: (gewichtete) Summe der Eingänge
- Aktivierung: Schwellwert-/Sigmoid-/tanh-funktion

Neuronales Netz

Neuronen über Ausgänge/gewichtete Eingänge verknüpft

Feed Forward: Anordnung in Schichten

Rekurrent: Mit Rückkopplung

Anpassung durch

Variation von Gewichten

Veränderung der Struktur

Anzahl der Neuronen

Verknüpfungen

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

3

Charakteristika typischer Anwendungen

Eingabe:

(reellwertige) Merkmalsvektoren.

Merkmale können auch korreliert sein.

Verrauschte Daten.

Ausgabe:

diskret/kontinuierlich

evtl. Vektor

Trainingsverfahren:

evtl. langwierig, komplex, wiederholte Versuche

Anwendung

Ausgabewert wird schnell berechnet

Schlechte Kommunizierbarkeit

„Subsymbolischer Ansatz“

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

4

Lernsystem Perzeptron

Aufgaben t :

Berechnung einer Funktion f für Eingabewerte $x_1, \dots, x_n$

Lernen:

Approximation von f

Hypothesen h (Suchraum):

Perzeptron = lineare Funktion

$$h(x_1, \dots, x_n) = w_0 + w_1 x_1 + \dots + w_n x_n$$

Beispiele D :

$$d = [x_1, \dots, x_n], f(x_1, \dots, x_n)$$

Analog für Klassifizierungsaufgaben t mit

$$h(x_1, \dots, x_n) = w_0 + w_1 x_1 + \dots + w_n x_n > \text{Schwellwert}$$

d.h. lineare Trennung

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

5

Lernverfahren Gradientenabstieg

Idee: Fehler bei Trainingsbeispielen minimieren

Summe der Fehlerquadrate:

$$E = \sum_{d \in D} (h(x_1^d, \dots, x_n^d) - f(x_1^d, \dots, x_n^d))^2$$

Gradientenabstieg:

Korrekturen der Gewichte:

$$w_i := w_i + \delta w_i \quad (\text{„Delta-Regel“})$$

in Richtung der größten Abnahme des Fehlers E

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

6

Lernverfahren Gradientenabstieg

0. Initialisierung:
 $w_0, \dots, w_n :=$ kleine zufällige Werte, Lernrate η festlegen.
- Iteration der Schritte 1-3
 mit jeweils neuer Hypothese h (korrigierte w_i)
 bis Abbruchbedingung (z.B. E unterhalb einer Schranke):
1. Für jedes Trainingsbeispiel $d = [x^d_1, \dots, x^d_n], f(x^d_1, \dots, x^d_n)$:
 Fehler $\delta^d := h(x^d_1, \dots, x^d_n) - f(x^d_1, \dots, x^d_n)$ berechnen
 2. Korrekturwerte der Gewichte w_i bestimmen:
 $\delta w_i := \eta \cdot \sum_{d \in D} \delta^d \cdot x^d_i$
 3. Gewichte korrigieren:
 $w_i := w_i + \delta w_i$

Best-mögliche Approximation von f , falls η geeignet gewählt.

H.D.Burkhard, HU Berlin
 Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
 Neuronale Netze (Ergänzungen)

7

Inkrementeller Gradientenabstieg

Gradientenabstieg annähern durch Einzelkorrekturen.
 Schnellere Konvergenz, Bessere Konvergenz bei lokalen Minima

- Initialisierung:
 $w_0, \dots, w_n :=$ kleine zufällige Werte, Lernrate η festlegen
- Iteration bis Abbruchbedingung:
 Für jedes Trainingsbeispiel $d = [x^d_1, \dots, x^d_n], f(x^d_1, \dots, x^d_n)$:
- a) Fehler $\delta^d := h(x^d_1, \dots, x^d_n) - f(x^d_1, \dots, x^d_n)$ berechnen
 - b) Korrekturwerte der Gewichte w_i bestimmen:
 $\delta w_i := \eta \cdot \delta^d \cdot x^d_i$ **DELTA-REGEL**
 - c) Gewichte korrigieren:
 $w_i := w_i + \delta w_i$
 - d) neue Hypothese h (korrigierte w_i)

H.D.Burkhard, HU Berlin
 Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
 Neuronale Netze (Ergänzungen)

8

Netze mit mehreren Schichten

Ausgabe o_u in jedem Neuron u gemäß seiner Eingabe berechnen
 Netz-Eingabe-Neuronen erhalten Eingabewerte $[x_1, \dots, x_n]$
 Innere „verdeckte“ Neuronen für Zwischenwerte
 Verbindung von Neuron v zu u mit Gewicht w_{uv}
 Matrix (w_{uv}) der Gewichte
 Propagierung von Aktivierungen längs der Verbindungen
 Netz-Ausgabe-Neuronen für Ausgabewerte $[t_1, \dots, t_k]$
 berechnete Funktion: $f(x_1, \dots, x_n) = [t_1, \dots, t_k]$

Lernverfahren: Backpropagation

Idee: Gradientenabstieg in den einzelnen Schichten
 Rückwärts-Propagierung von Fehlern
 Differenzierbarkeit: Sigmoid-Funktion statt Schwellwert

H.D.Burkhard, HU Berlin
 Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
 Neuronale Netze (Ergänzungen)

9

Backpropagation

- Initialisierung: Netzstruktur vorgeben,
 $(w_{uv}) :=$ kleine zufällige Werte, Lernrate η festlegen
- Iteration Schritt bis Abbruchbedingung erfüllt:
- Für jedes Trainingsbeispiel $d = [x_1, \dots, x_n], [t_1, \dots, t_k]$
- a) Vorwärtspropagierung:
 Für jedes Neuron u die Ausgabe o_u berechnen.
 - b) Rückwärtspropagierung des Fehlers
 für Ausgabe-Neuronen u
 $\delta_u := o_u \cdot (1 - o_u) \cdot (t_u - o_u)$
 für innere Neuronen v (schrittweise „nach rückwärts“)
 $\delta_v := o_v \cdot (1 - o_v) \cdot \sum_u w_{uv} \cdot \delta_u$
 - c) Korrekturwerte der Gewichte w_i bestimmen:
 $\delta w_{uv} := \eta \cdot \delta_u \cdot x_{uv}$
 - d) Gewichte korrigieren:
 $w_{uv} := w_{uv} + \delta w_{uv}$
 - e) neue Hypothese h (korrigierte w_i)

Weitere Probleme

Overfitting:
 Anpassung an spezielle Beispiele zulasten Generalisierung
 Gegenstrategie:
 Beispielmenge D aufteilen in
 Trainingsmenge D_T , Validierungsmenge D_V
 Abbruch-Kriterium beziehen auf D_V
 Kreuzvalidierung

Simulated Annealing:
 Wert für Lernrate η allmählich vermindern
 - als Strategie gegen lokale Minima

H.D.Burkhard, HU Berlin
 Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
 Neuronale Netze (Ergänzungen)

11

Weitere Probleme

Strukturveränderung
 Passende Anzahl wesentlich für Erfolg

Neuronen hinzunehmen (bei schlechter Konvergenz)

Neuronen entfernen (falls ohne Einfluß auf Konvergenz)

Genetische Verfahren zur Struktur-Variation

Bedeutungstragende Neuronen
 („Großmutter-Neuron“)

H.D.Burkhard, HU Berlin
 Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
 Neuronale Netze (Ergänzungen)

12

Suchraum, Inductive Bias

Suchraum:

Euklidischer Raum mit n Dimensionen
(n = Zahl der Gewichte)

Suchverfahren: Gradientenabstieg bzgl. Fehlerfunktion

Leistungsfähigkeit: Approximation beliebiger Funktionen

Inductive Bias:

„Stetige Interpolation“ von Zwischenwerten der gelernten Funktion

(analog für Klassenzugehörigkeit)

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

13

Zusammenfassung

Subsymbolische Verarbeitung

Approximation reellwertiger Funktionen

Robust bzgl. Rauschen in Trainingsdaten

Robust bzgl. Ausfall von Neuronen

Backpropagation benutzt Gradientenabstieg als universelles Verfahren für Hypothesen mit reellwertigen Parametern

Konvergenz zu (lokalem) Minimum der Fehlerfunktion

Standardverfahren: exakter Gradient, größere Schrittweite

Inkrementell: vermeidet besser lokale Minima

Overfitting: Kreuzvalidierung

Inductive Bias: Stetige Interpolation

H.D.Burkhard, HU Berlin
Sommer-Semester 2007

Vorlesung MMKI: Maschinelles Lernen
Neuronale Netze (Ergänzungen)

14