Stand der Kunst 1 in Künstlicher Intelligenz

Schach, Neuronale Netze, Vision, Robotik

Benjamin Teuber

Fachbereich Informatik
Universität Hamburg

28.06.2005 / PS Denkmaschinen

Gliederung

Drew McDermott

- KI-Forscher
- Anspruch: Mind-Body-Problem erklären
- seit 1976 an der Yale-Universität
- 1973-76 am MIT
- politsche Haltung: starker Bush-Kritiker
- arbeitet derzeit an Agentensysteme, Wissensrepräsentationen etc.

Missverständnis über KI

- Irrglaube: KI hätte mit Intelligenz zu tun
- McDermott: “KI” ist ein unglücklicher Name
  - Anwendung von Informatik auf Kognitionswissenschaften
  - beschäftigt sich mit “intelligenten” Problemen, z.B. Schach

It would be better, therefore, if AI had a different name, such as *cognitive informatics*. (...) AI will have a lot to say about what role computation plays in thinking, but almost nothing to say about intelligence.

GOFAI
Good Old Fashioned Artificial Intelligence

- Ansatz: Logische Verarbeitung von Symbolen
- angenommene Trennung zwischen “Low- und High-Level-Aktivitäten” im Gehirn
  - Algorithmen, die ein “intelligentes” Probleme lösen, sind auf andere solcher Aufgaben Übertragbar
  - “Low-Level”-Aktivitäten wie z.B. Laufen können “straightforward” mit Hilfe der Kontrolltheorie gebaut werden
- beide Annahmen haben sich nicht erfüllt
  - es gibt keinen “General Problem Solver”
  - keine prinzipiellen Unterschiede zwischen Low-Level und High-Level im Hirn
  - erst recht keinen Platz der “Puren Intelligenz”

Heutige Einstellung

- bescheidener
- man versucht nicht mehr, das gesamte menschliche Denken am Stück zu erklären
- Beschränkung auf Teilgebiete, die man perfektioniert

People tend to underestimate the difficulty of achieving true machine intelligence. Once they've seen a few examples, they start drawing schematic diagrams of the mind, with boxes labelled "perception" and "central executive". (...) It doesn't take long for them to convince themselves that intelligence is a simple thing, just one step beyond Windows 98.

Gliederung

Schach

- eines der ersten Anwendungsfelder der KI
- im Computer einfach zu repräsentieren
- gilt als anspruchsvoll
- keine offensichtlichen Algorithmen
- erster Ansatz: abstrakte Symbolverarbeitung, Pläne etc.
- mit der Zeit immer spezialisiertere Programme
- inzwischen kein Anspruch mehr, menschliches Denken abzubilden

Wie funktionieren Schachprogramme?

- Eigenschaften von Schach
  - keine Unsicherheit (wie z.B. Kartenspiele)
  - keine Zufälligkeit ($\rightarrow$ Backgammon)
  - $\Rightarrow$ einfache Repräsentation aller möglichen Fortsetzungen
  - endlicher Spielbaum
- Grundidee von Shannon und Turing
- baut auf Spieltheorie-Arbeiten von von Neumann und Morgenstern auf

MiniMax-Algorithmus

- baue einen Baum aller möglichen Spiele auf
  - Knoten $\hat{=}$ Stellung
  - Kinder $\hat{=}$ mit einem Zug erreichbare Stellungen
  - Blätter $\hat{=}$ Endstellungen
- ordne allen Blättern die Werte zu:
  - 1 $\hat{=}$ gewonnen
  - 0 $\hat{=}$ unentschieden
  - -1 $\hat{=}$ verloren
- nun rekursiv nach oben Werte vergeben
  - denn: Jeder Spieler macht den Zug mit dem besten Ergebnis
  - zwei Fälle:
  - Spieler dran: $wert(knoten) := \max\{wert(k)\}, k \in kinder(knoten)$
  - Gegner dran: $wert(knoten) := \min\{wert(k)\}, k \in kinder(knoten)$
- im Wurzelknoten: Mache Zug, der zur besten Stellung führt

Minimax: Ein sicheres Gewinnrezept?

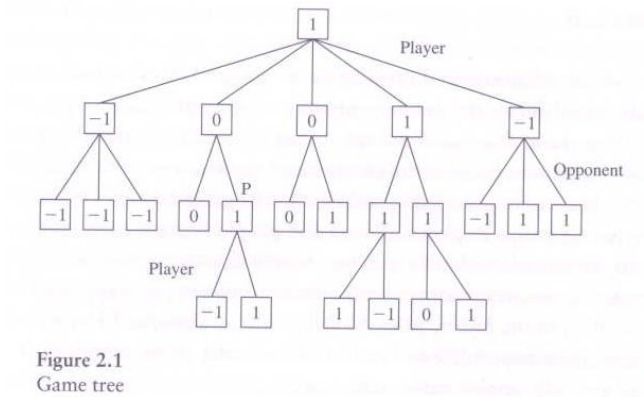

Figure 2.1
Game tree

- Minimax terminiert und findet in endlicher Zeit garantiert den besten Zug
- für Menschen allerdings unbrauchbar - sie können nicht so viele Stellungen bearbeiten
- “Aber doch sicher Computer?”

Eine simple Rechnung

- Zugmöglichkeiten:
  - am Anfang 12 Züge möglich
  - im Mittelspiel meist mehr
  - im Endspiel weniger
  - Annahme: 10 Möglichkeiten für jeden Zug
- Durchschnittstiefe: z.B. 100 Halbzüge (sicher nicht übertrieben)
- $\Rightarrow$Baum hat 10^{100} Stellungen!!!
- zum Vergleich:
- geschätzte Anzahl Atome im Universum: um die 10^{78}
- also: Computer können nur einen kleinen Ausschnitt des Baums aufbauen

Wie Programme wirklich arbeiten

- Teilbaum, z.B. 5 Züge tief, wird aufgebaut
- Endstellungen wie gewohnt mit -1, 0, 1 markieren
- Bewertungsfunktion
  - schätzt Blatt-Stellungen
  - ergibt reelle Werte zwischen -1 und 1
  - z.B. 0,9 für eine sehr gute Position
  - Grundprinzip: Material beider Seiten vergleichen
  - Erweiterungen möglich, z.B. Anzahl möglicher Züge
- dann wieder Minimax...

Fine-Tuning für Schachprogramme

- Endspielbibliothek
  - aus allen Mattpositionen zurückanalysieren und speichern
  - Stand: Für 5 Figuren auf dem Brett ist Ergebnis gespeichert!
- Eröffnungsbibliothek
- erweiterte Algorithmen: Alpha-Beta, Erwartungsfenster, Ruhesuche, Zobrist-Hashing, Null-Move uvm.
- Ergebnis:
  - kaum Wissen über menschliche Spielweise fließt in Schachprogramme ein
  - aber: Deep Blue schlägt Kasparov 1997
  - einige Programme mit symbolischer KI - keine Erfolge
  - Schach analog zur generellen KI-Entwicklung

Sind Schachprogramme intelligent?

- Kritiker:
  - Programme sind nicht intelligent
  - ganz andere Verfahren als beim Menschen
  - Computer berechnen nur, Menschen besitzen Erkenntnis und Urteilsvermögen
- McDermott:
  - Gehirn kann nicht als digitaler Computer verstanden werden
  - aber:
    - “Erkenntnis” und “Urteilsvermögen” mystifizieren Gehirnleistung, ohne das etwas konkretes dahintersteckt
    - Neuronen führen Berechnungen durch
    - entscheidend für Denken mit Neuronen ist nicht chemische Implementation, sondern Information & ihre Verarbeitung

Neuronale Netze

- starke Vereinfachung der Gehirn-Arbeitsweise
- überwinden einige KI-Einschränkungen
- Graphendarstellung
  - Knoten $\hat{=}$ Neuron
  - Kante $\hat{=}$ Synapse
- Output eines Neurons wird anderen als Input gegeben
- einige Ein- oder Ausgabeneuronen
- Synapse hat Gewichtungsfaktor
- Zahlenwerte typischerweise im Bereich $[-1, 1]$
- alle Inputs werden gewichtet aufsummiert
- nicht-lineare Aktivierungsfuntion, z.B. $sgn(x)$ oder $tanh(x)$

Neuronale Netze (2)

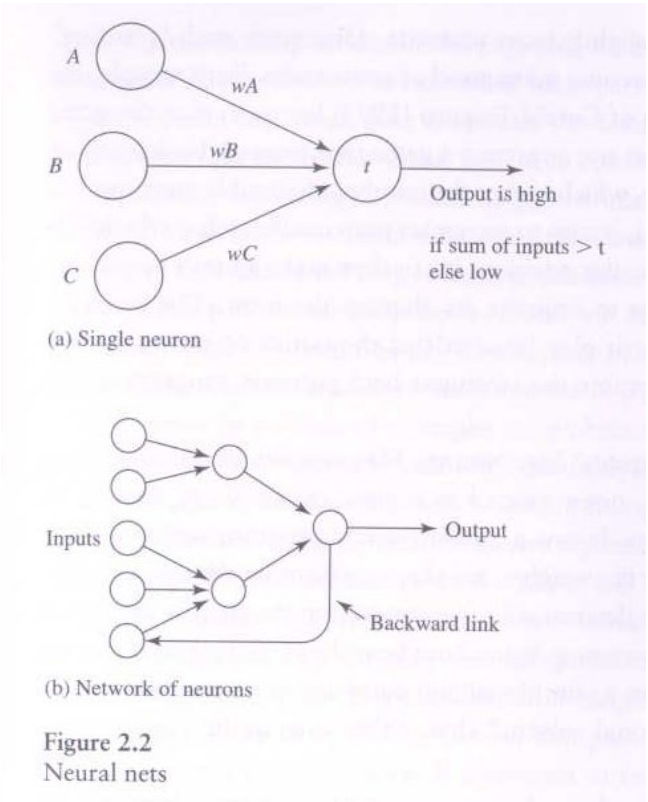

Figure 2.2
Neural nets

- Variationen:
  - getrennte Schichten von Neuronen, die nacheinander bearbeitet werden → mit Matrizenrechnung erfassbar
  - Zyklen erlaubt → Rückkopplungen wie im Gehirn

- verbreitete Meinung: Neuronale Netze benutzen ein komplett anderes Berechnungsmodell als digitale Computer
- aber: kann auch auf digitalem Computer simuliert werden

Lernfähigkeit Neuronaler Netze

- ähnlich dem natürlichen lernen
- anpassung der Synapsenstärken und Membransensitivität
- ein Lernalgorithmus: Backpropagation
  - Netzausgabe mit gewünschtem Ergebnis vergleichen
  - Kantengewichte leicht ändern,so dass der Fehler sich verkleinert
  - $\rightarrow$ mehrdimensionale Optimierung - Analysis
  - lernt nicht selbstständig, sondern über Feedback

Backgammon

- im Gegensatz zu Schach mit Würfeln
- $\Rightarrow$Minimax-Suche funktioniert nicht gut, da
  - größerer Variantenbaum
  - nur Gewinnwahrscheinlichkeiten, keine Sicherheit
- TD-gammon
  - Backgammon-Programm von Tesauro
  - nicht Spielbaum, sondern Stellungsbewertung durch ein neuronales Netz:
    - Eingabe: Stellung
    - Ausgabe: Gewinnwahrscheinlichkeit

TD-gammon

- Training:
  - hat millionenfach gegen sich selbst gespielt
  - nach Partieende Anpassung der Bewertungsfunktion:
    - Gewinnseite→bessere Bewertung
    - Verlustseite→schlechtere
- Ergebnis: wurde stärkstes Programm der Welt!

Weltklassespieler Kit Woolsey über TD-gammon:

There is no question that its positional judgment is far better than mine. Only on small technical areas can I claim a definite advantage over it...

TD-gammon vs. Deep-Blue

- beide auf low-level Ähnlich -Assembler-Befehle
- trotzdem komplett unterschiedliche Herangehensweise
- $\Rightarrow$ komplementäre Fähigkeiten
- also:
  Eigenschaften von Prozessen, insbesondere Einsicht und Erkenntnis, lassen sich nicht aus den “mechanischen” Schritten beurteilen, die sie benutzen

Computer-Vision

- Biologische Seh-Systeme können gut bewegte Objekte erkennen
- Prinzip:
  - Helligkeitsveränderung benutzen, um Geschwindigkeitsvektoren aller Punkte zu berechnen
  - $I(P,t) = I(P - vt, 0)$
- Intensität kann nur an endlich vielen Punkten gemessen werden

- Natur:
  - verteilte Rezeptoren
  - Bewegungserkennungs-Neuronen
  - Konnektionistisch
  - evolutionär entwickelt

- Computer:
  - Gleichmäßiges Raster
  - Zahlenarrays
  - Von-Neumann-Architektur
  - geplant entwickelt

Robotik

- Problem: Bewegung über Sensoren kontrollieren und anpassen
- keine festen Bewegungsbahnen wie bei Automobilen, sondern verschiedene Gelenke
- es werden genaue Raum-Informationen benötigt

Erstellung von Depth Maps

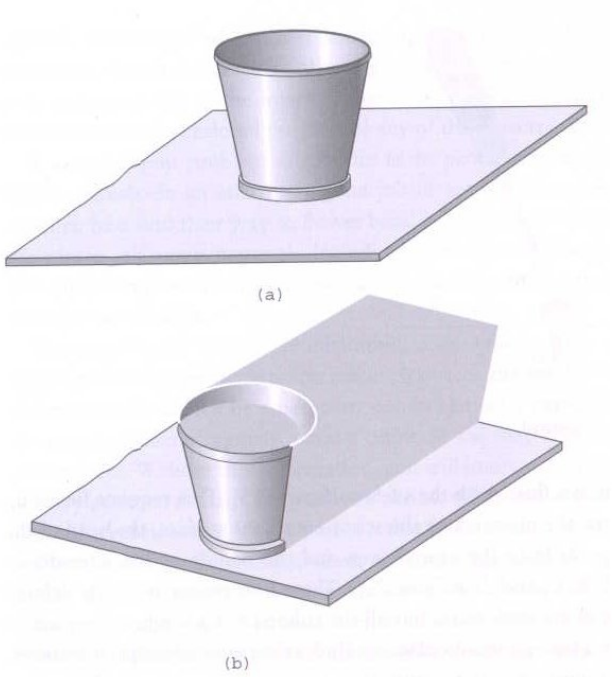

Figure 2.4
(a) Bucket. (b) Depth map from (a) (seen from the side)

- Aufgabe: aus Sensorinformationen 3d-Repräsentation berechnen
- Ansätze:
  - 2 Augen, die leicht auseinander liegen
  - Textur und Schatten analysieren
  - Laser-Range-Finding
- Problem: keine vollständige Repräsentation
  → Gedächtnis nötig

Map-Learning

- Tiere können sich gut orientieren:
- Bienen finden Blumen
- Vögel ziehen nach Süden
- Ziel: Robotern ähnliche Fähigkeiten zu geben

- was ist mit “Map” gemeint?
- Vorstellung: Stück Papier mit schematischer Weltrepräsentation
- wichtig: Erkennen, wo auf der Karte man gerade ist
- z.B. Straßennamen, markante Punkte (Baum, Hügel, etc.)

Dead Reckoning

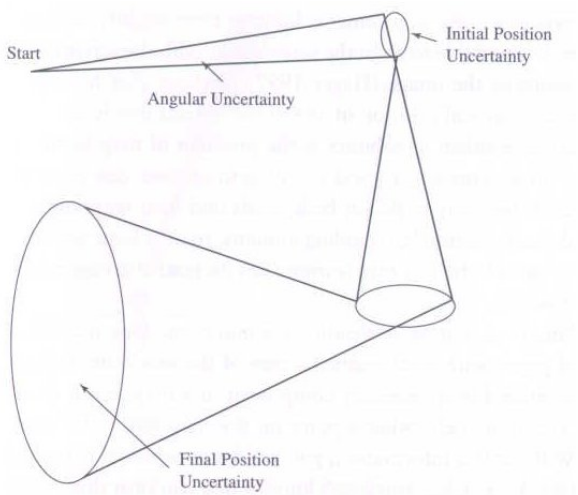

Figure 2.6
Positional uncertainty due to angular uncertainty

- Begriff kommt aus Seefahrt
- Navigation durch Protokollieren der Bewegungen
- ohne Orientierungspunkte, z.B. Sterne
- wird von Tieren benutzt
- Anwendung bei Robotern problematisch
  - Kleine Messfehler verursachen große positionale Unsicherheit
  - $\Rightarrow$ Roboter muss zusätzlich Orientierungspunkte benutzen

Sonar vs. Kameras

- Sonar:
  - Billig
  - Viel verwendet
  - Ungenau: 30 Grad Unsicherheit
  - 1d-Ring, z.B. 16 Strahlen a 1 Byte
- Kamera:
  - 2d-Bild
  - z.B. 10.000 S/W Pixel a 1 Byte
- $\Rightarrow$ Kamera liefert mehr Informationen

Probleme beim Wiedererkennen

- Problem: Der selbe Ort erscheint unterschiedlich
  - Kameraposition etwas anders
  - Objekte leicht verändert
  - Lichtverhältnisse
- Auswirkungen von Kamerabewegungen auf Pixel analysieren

Algorithmus von Hemant Tagare

- Idee: Bild 2 als zufällige “Störung” von Bild 1 ansehen
- Für jeden Pixel aus Bild 2
  - Betrachte umliegende Region aus Bild 1
  - Berechne Farbschwankungen und -intervall
  - Fällt Pixelwert in Farbinterval?
    - Nein→Markiere Pixel als Fehlpixel(“outlier”)
    - Ja →Speichere Abstand von Quell- und Zielpixel
- Ergibt zwei Charakteristische Werte:
  - Anzahl Fehlpixel
  - Durchschnittsabstand bei nicht-Fehlpixeln
- Beide niedrig⇒Hinweis, dass es derselbe Ausschnitt sein könnte

Wahrscheinlichkeits-Karten

- Keine sicheren Informationen, daher bietet sich W.-Rechnung an
- Karte als Zuordnung Punkt$\rightarrow$W. möglicher Landmarken (“landmarks”)
- Nullmarke $\hat{=}$ nichts besonderes

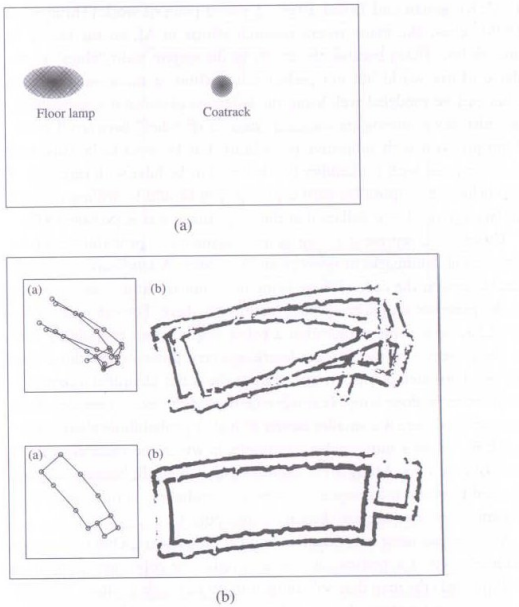

Figure 2.9
(a) Simple probabilistic map. (b) More complex map (from Thrun et al. 1998)

- Lokalisation
  - Formal: finde Ort X, so dass $P(\text{Rotober befindet sich in } X \mid \text{Karte } M \wedge \text{ Beobachtungen } D)$ maximal wird
  - Beobachtungen: z.B. gesehene Landmarken, DR-Ergebnis etc.

Karten erstellen

- finde Karte M, so dass $P(M|\text{Beobachtungen } D)$ maximal
  - Umformung $P(M|D)$
  - $= \frac{P(D|M)P(M)}{P(D)}$ (Bayes-Theorem)
  - $P(D|M)$ einigermaßen zu berechnen
  - $P(M)$ kann Axiome enthalten, z.B. “es ist Unwahrscheinlich, dass ein Haus in einem Fluss steht” - meist wird aber nur wenig vorausgesetzt
  - $P(D)$ nicht von M abhängig, also hier irrelevant
- Test
  - Roboter wurde 15 Minuten herumbewegt, danach Berechnung der wahrscheinlichsten Karte
  - Ergebnis: Bis auf cm genau!

Gliederung

Fazit

- Verschiedene zentrale Algorithmen der KI
  - Baumsuche (Minimax)
  - Neuronale Netze
  - Wahrscheinlichkeitsanalysen
- Erfolge der KI
  - Spiele: Schach, Backgammon uvm.
    aber: Beim asiatischen Brettspiel Go versagt jeder Ansatz bisher...
  - 3d-Repräsentationen & Karten aus Bildern erstellen

Behauptung

Das menschliche Gehirn basiert auf neuronaler Informationsverarbeitung, deren chemische Umsetzung für das Ergebnis irrelevant ist. Diese Form der Informationsverarbeitung ließe sich auf einem digitalen Computer nachbilden, woraufhin das Ergebnis nicht Qualitativ schlechter wäre als das menschliche Gehirn.

Was ist Go?

- das älteste Brettspiel der Welt - geschätzt auf ca. 2000 v.Chr.
- in China entstanden
- in Japan perfektioniert
- heute von Korea dominiert
- einfache Regeln - unmöglich zu meistern
- benötigt beim Menschen Analysefähigkeit und Intuition gleichermaßen

Go als KI-Testfeld

- in Go-Programmierung wurde inzwischen mehr Zeit investiert als in Schach
- dennoch: kein Programm kommt gegen einen halbwegs fortgeschrittenen Menschen an!
- Probleme der Go-Programmierung
  - gigantischer Variantenbaum: $4{,}36 * 10^{170}$ mögliche Stellungen
    Vergleich: bei Schach "nur" 10^{43}
  - keine einfache Bewertungsfunktion
  - Kontextabhängigkeit ähnlich bei Sprachverarbeitung

Go-KI-These

Um Go zu spielen, benötigt man "echte" Intelligenz, nämlich ein dynamisches Zusammenspiel aus Intuition und Analysefähigkeit. Ob man dies durch Kombination der Prinzipien Baumsuche und Neuronale Netze erreichen kann, bleibt noch offen...

Quellen


McDermott, Drew
Mind and Mechanism
The MIT Press, 2001


Steinwerder, Dieter; Friedel, Frederic A.
Schach am PC
Markt & Technik, Buch- und Softwareverl., 1995


Zell, Andreas
Simulation neuronaler Netze
R.Oldenbourg Verlag, 1997

Quellen (2)

- http://de.wikipedia.org/wiki/Go_ (Brettspiel)
- http://de.wikipedia.org/wiki/Roboter
- http://cs-www.cs.yale.edu/homes/dvm
- http://www.cs.yale.edu/people/mcdermott.html