

Grundlagen der Künstlichen Intelligenz

Sebastian Iwanowski
FH Wedel

Kap. 4:
KI-Architektur: Wissensbasierte Systeme

4.3: Fallbasierte Wissensverarbeitung

2. Fallbasierte Diagnose

Eingabe in die Wissensbasis:

- Fälle mit vollständigem Symptomvektor und zugehörigen Fehlern (eindeutig klassifiziert)

a) KI-klassisch, mit Ähnlichkeitsmaß:

- Ähnlichkeitsmaß für unvollständige Symptomvektoren (gewichtet häufig zwischen den verschiedenen Symptomtypen)

Struktur der Wissensbasis:

- Punkte im Vektorraum
- Ähnlichkeitsmetrik

Arbeit der Problemlösungsmaschine:

- Finde zu gegebenem neuen Vektor den ähnlichsten Symptomvektor in der Wissensbasis.
- Weise dem neuen Fall dieselben Fehler zu, die dem Referenzvektor in der Wissensbasis zugeordnet sind (u.U. mit Angabe der Treffsicherheit).

2. Fallbasierte Diagnose

Eingabe in die Wissensbasis:

- Fälle mit vollständigem Symptomvektor und zugehörigen Fehlern (eindeutig klassifiziert)

b) mit Neuronalen Netzen:

- Neuronales Netz mit Eingabeschicht (für Symptomvektor) und Ausgabeschicht (für Fehler) sowie (optional) Zwischenschicht aus Knoten und Kanten, jeweils mit veränderbaren Gewichten.

Struktur der Wissensbasis:

- Punkte im Vektorraum
- Neuronales Netz mit konkret eingestellten Gewichten (abhängig von Symptomvektoren und zugehörigen Fehlern der Eingabe)

Arbeit der Problemlösungsmaschine:

- Lege gegebenen neuen Symptomvektor an die Eingabeschicht des neuronalen Netzes an.
- Lies die zugeordneten Fehler aus der Ausgabeschicht ab.

Fallbasierte Problemlösung (CBR)

Verallgemeinerung der fallbasierten Diagnose auf beliebige fallbasierte Auswertungsstrategien:

Prinzip:

- Gegeben seien Fälle in Form von Vektoren (*vollständige Symptomvektoren*): Diese werden “gelernt” und bilden die Wissensbasis.
- Gegeben seien neue Vektoren, in denen nicht alle Komponenten bekannt sind (*unvollständige Symptomvektoren*): Diese sind zu klassifizieren.
- Ordne den unbekanntem Komponenten der neuen Vektoren Werte zu.

Arbeit des Problemlösers (einfache Variante):

- Finde zu gegebenem neuen Vektor den “ähnlichsten” Symptomvektor in der Wissensbasis.
- Weise den unbekanntem Komponenten des neuen Vektors dieselben Werte zu, die den entsprechenden Komponenten des Referenzvektors in der Wissensbasis zugeordnet sind.

Dieses Verfahren ist nur sinnvoll, wenn die unbekanntem Werte aus einem diskreten (besser: endlichen) Wertebereich kommen !

Fallbasierte Problemlösung (CBR)

Verbesserung für kontinuierliche Wertebereiche:

Arbeit des Problemlösers (bessere Variante):

- Weise den unbekannt Komponenten des neuen Vektors Werte zu, die zwischen den Werten der entsprechenden Komponenten von "in der Nähe liegenden" Vektoren der Wissensbasis liegen.

Andere (mathematische) Formulierung dieser Methode:

- Fasse die unbekannt Werte der neuen Vektoren als Funktionswerte der bekannten auf: Finde eine stetige Funktion, von der alle Vektoren der Wissensbasis eine Lösung sind.
- Weise den neuen Vektoren die Funktionswerte der bekannten Komponenten zu.

Frage: Wie findet man die Funktion zu einer gegebenen Menge von Referenzvektoren ?

Antwort:

- Gib eine Funktionsklasse vor, innerhalb der sich die Funktionen durch Parameter unterscheiden.
- Bestimme die Parameter als Lösung eines Gleichungssystems aus den bekannten Symptomvektoren.

Fallbasierte Problemlösung (CBR)

Parameterbestimmung in Funktionsklassen (Regression):

Lineare Regression:

- Finde die Gewichte in linearen Funktionsgleichungen der Form $f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n W_i x_i$

Verallgemeinerung:

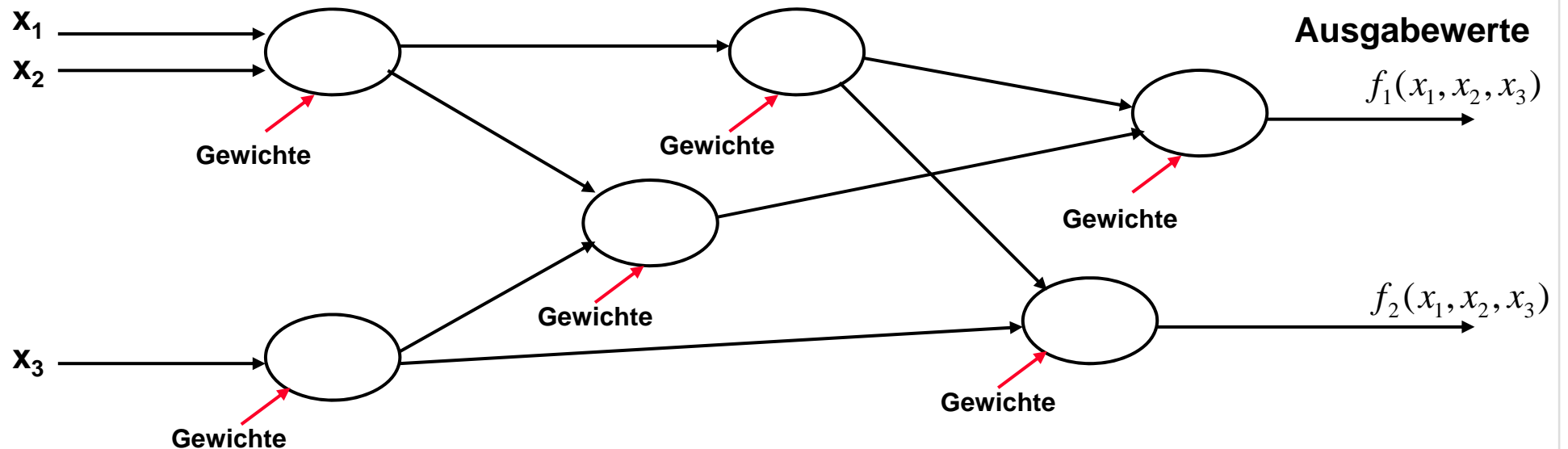
1. Finde die Gewichte in einem linearen Gleichungssystem.
 2. Finde die Gewichte in Gleichungssystemen höherer Ordnung.
 3. Finde die Gewichte in parametrisierten Ungleichungssystemen.
- **Der fallbasierte Ansatz ist gerade für Systeme gedacht, die man *nicht* leicht modellieren kann.**
 - **Daher bringt der Ansatz mit höheren Gleichungssystemen nicht viel.**
 - **Besser ist es, mit vielen nur lose miteinander gekoppelten Gleichungssystemen zu arbeiten und das unsichere Wissen zu verteilen.**

Fallbasierte Problemlösung (CBR)

Idee der neuronalen Netze:

Gegeben sei eine mehrdimensionale Funktion f (Schreibweise: $f_i(x_1, x_2, \dots, x_n)$)

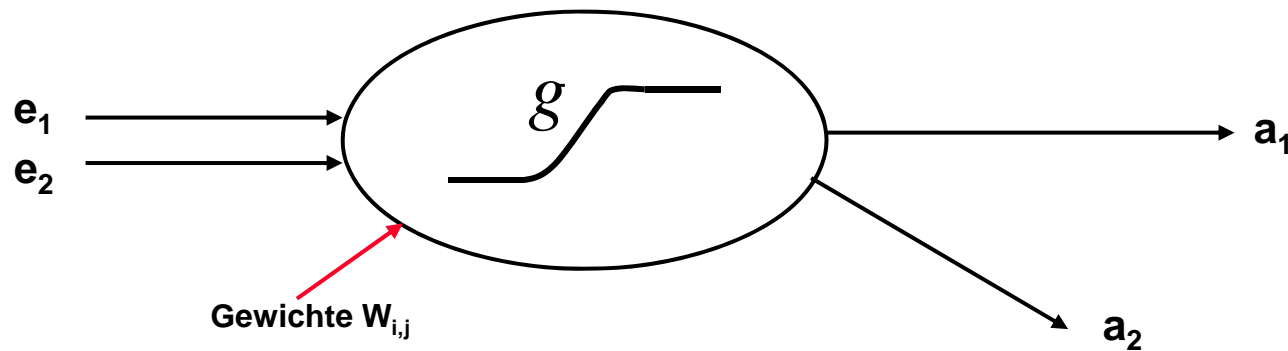
Eingabewerte



- Die Gewichte können voreingestellt werden, werden aber an die gelernten Beispiele angepasst.
- Neue Funktionswerte werden durch Durchlauf des Netzes ausgerechnet.

Fallbasierte Problemlösung (CBR)

Funktionsweise eines einzelnen Neurons:



$$a_i(e_1, e_2, \dots, e_n) = g \left(\sum_{j=1}^n W_{i,j} e_j \right)$$

- g ist eine verallgemeinerte Schwellenwertfunktion, die für alle Ausgaben desselben Neurons gleich ist.

Fallbasierte Problemlösung (CBR)

Verschiedene Stufen neuronaler Netze:

Neuronale Netze ohne Zwischenschichten:

- Es gibt Neuronen in einer ersten Schicht, die mit der Eingabe verbunden sind und das Ergebnis an Neuronen einer zweiten Schicht, der Ausgabeschicht leiten, welche mit der Ausgabe verbunden sind.

Neuronale Netze mit Zwischenschichten

- Eingabe- und Ausgabeschicht sind über weitere “verborgene Schichten von Neuronen verbunden.

Neuronale Netze mit Rückkopplung:

- Ausbildung eines “Gedächtnisses”

Fallbasierte Problemlösung (CBR)

Was ist der entscheidende Unterschied zwischen neuronalen Netzen und „klassischen“ fallbasierten Systemen ?

- **In neuronalen Netzen ist das Wissen über die gelernten Fälle **verteilt**.**

Theoretische Vorteile der Verteilung:

- Willkürlichkeit des Funktionsklassenansatzes spielt nicht so eine große Rolle
- Undurchschaubare Fälle bekommen einen undurchschaubaren Ansatz:
Der verteilte Ansatz “reguliert sich selbst”.

In der Praxis hat sich gezeigt:

- Für gut funktionierende neuronale Netze benötigt man weniger Lernbeispiele als in klassischen fallbasierten Systemen.
- Neuronale Netze liefern bessere Resultate bei der Klassifizierung.

Neuronale Netze und KI

Sind neuronale Netze wissensbasiert ?

Sind neuronale Netze Expertensysteme ?

Was verdient den Namen „Künstliche Intelligenz“ ?

Zusammenfassung: Fallbasiertes Schließen

Vorteile und Nachteile:

- **Das Verfahren ist einfach.**
 - Die Diagnose der Laufzeitkomponente ist schnell.
 - Der Wissenserwerb ist leicht automatisierbar.
 - Die Wissensbasis kann nur für Systeme erstellt werden, für die Erfahrungswissen vorhanden ist.
 - Die Wissensbasis nimmt sehr viel Speicherplatz ein (nur bei Ähnlichkeitsmetrik).

Zusammenfassung: Fallbasiertes Schließen

Vorteile und Nachteile:

- **Die Wissensbasis enthält überhaupt keine andere Strukturinformation als das Ähnlichkeitsmaß bzw. das NN.**
 - Alle Anwendungsbereiche sind gleich geeignet.
 - Die Problemlösungsmaschine kann unverändert für alle Anwendungsbereiche eingesetzt werden.
 - Die Wissensbasis ist schon bei einer kleinen Änderung des Systems nicht mehr zuverlässig einsetzbar.
 - Das Ähnlichkeitsmaß bzw. Neuronale Netz ist häufig willkürlich.
 - Jede Laufzeitdiagnose kann falsch sein.
 - Die Diagnose ist überhaupt nicht nachvollziehbar (nur für Neuronale Netze).