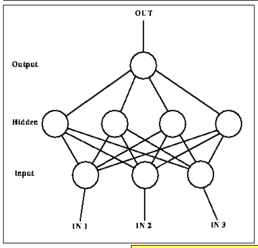
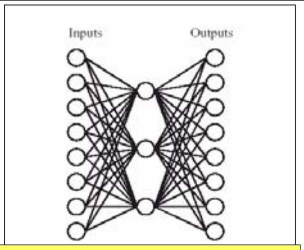
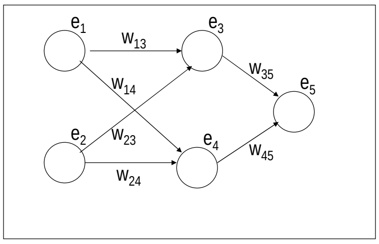
„Massive Parallelität“: Neuronale Netze





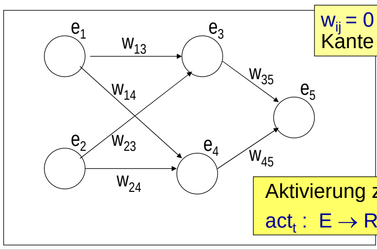
- Knoten: Neuronen
  Neuronen können erregt („aktiviert“) sein
- Kanten: Übertragung von Aktivierungen
  an Nachbar-Neuronen

„Massive Parallelität“: Neuronale Netze



- Kanten sind gewichtet
- Übertragene Aktivierung abhängig von Gewicht w_{ij}

„Massive Parallelität“: Neuronale Netze



$w_{ij} = 0$, falls keine Kante zwischen e_i und e_j

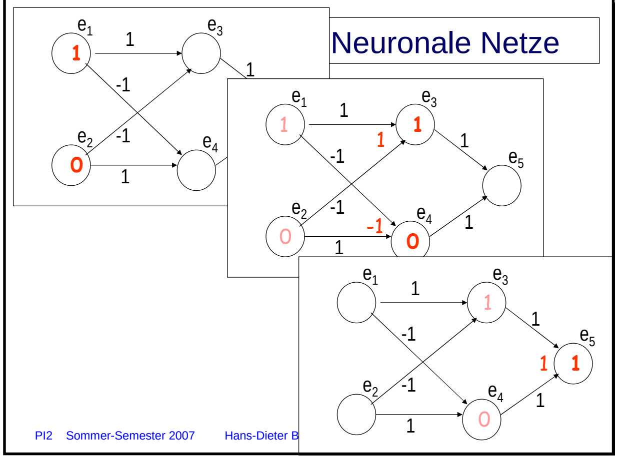
Aktivierung zum Zeitpunkt t :
$act_t : E \rightarrow R$

Einfache Propagierung von Aktivierungen z.B. gemäß:

$act_{t+1}(e_i) = 1$,
falls $w_{1i} * act_t(e_1) + w_{21} * act_t(e_2) + ... + w_{5i} * act_t(e_5) > 0$

$act_{t+1}(e_i) = 0$, sonst

Neuronale Netze



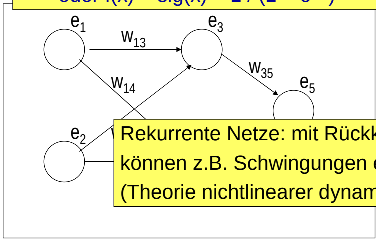
„Massive Parallelität“: Neuronale Netze

Komplexere Propagierung von Aktivierungen:

$act_{t+1}(e_i) = f(w_{1i} * act_t(e_1) + w_{21} * act_t(e_2) + ... + w_{5i} * act_t(e_5))$

z.B. mit $f(x) = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$

oder $f(x) = sig(x) = 1 / (1 + e^{-x})$

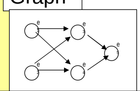


Rekurrente Netze: mit Rückkopplungen, können z.B. Schwingungen erzeugen (Theorie nichtlinearer dynamischer Systeme)

Statische/dynamische Modelle

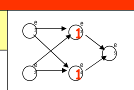
- Beschreibung statischer Zusammenhänge
  - Semantisches Netz
  - Klassendiagramm
  - Zustandsdiagramm (als Struktur)
  - Neuronales Netz (als Struktur)

Modell: Graph



- Beschreibung dynamischer Abläufe
  - Aktueller Zustand im Zustandsdiagramm
  - Aktivierung im Neuronalen Netz

Unterscheiden!



Modell:
Wechselnde Beschriftungen des Graphen

Erfordert Angaben für Übergänge (Transitionen) zwischen Zuständen, Aktivierungen usw.

Darstellungsformen für Graphen

Adjazenz-Matrix:

- Ausgangs-Knoten v als Zeilen
- Eingangs-Knoten v′ als Spalten

Matrix-Elemente:

$m_{v,v'} = 1$, falls $[v,v'] \in E$
$m_{v,v'} = 0$, falls $[v,v'] \notin E$

bzw. bei Kanten-Beschriftungen auch:

$m_{v,v'} = \beta([v,v'])$

dabei spezieller Wert für $[v,v'] \notin E$, z.B. $\beta([v,v'])=0$

Darstellungsformen

Inzidenz-Matrix:

- Knoten v als Zeilen
- Kanten e als Spalten

Matrix-Elemente:

$m_{v,e} = +1$, falls $e= [v, v'] \in E$
$m_{v,e} = -1$, falls $e= [v', v] \in E$
$m_{v,e} = 0$, sonst

Weitere Definitionen zu Graphen

$G' = [V',E']$ ist Teilgraph von $G=[V,E]$,
falls $V' \subseteq V$ und $E' \subseteq E$ mit $E' \subseteq V' \times V'$

Einschränkung von $G=[V,E]$ auf $V' \subseteq V$:
$G|V' =_{Def} [V', E \cap V' \times V']$

Für $e = [v,v'] \in E$: $\text{source}(e) =_{Def} v$, $\text{target}(e) =_{Def} v'$

Eingangsvalenz (fan-in), Ausgangsvalenz (fan-out)

$\text{fan-in}(v) =_{Def} \text{card} (\{ e \mid \exists v' \in V: e= [v',v] \in E \})$
$= \text{card} (\{ e \in E \mid \text{target}(e) = v \})$
$\text{fan-out}(v) =_{Def} \text{card} (\{ e \mid \exists v' \in V: e= [v,v'] \in E \})$
$= \text{card} (\{ e \in E \mid \text{source}(e) = v \})$

Weg (Pfad) in einem Graphen

Gerichteter Weg

Als Folge von Knoten

$p = v_0, \ldots, v_n$ mit $\forall i (i \in \{ 1,\ldots,n\} \rightarrow [v_{i-1},v_i] \in E)$
n ist die *Länge* des Weges ($n \geq 0$)
v_o = Anfang(p) v_n = Ende(p)

Als Folge von Kanten

$p = e_1, \ldots, e_n$ mit $\forall i(i \in \{1,\ldots,n-1\} \rightarrow \text{target}(e_i)=\text{source}(e_{i+1}))$

Ungerichteter Weg:

$p = v_0, \ldots, v_n$ mit $\forall i(i \in \{1,\ldots,n\} \rightarrow [v_{i-1},v_i] \in E \vee [v_i,v_{i-1}] \in E)$

Weg (Pfad) in einem Graphen

Einfacher Weg:

$p = v_0, \ldots, v_n$ mit $\forall i \, \forall j \in \{ 0,\ldots,n \}: v_i = v_j \rightarrow i = j$

Masche:

Zwei unterschiedliche einfache Wege p und q der Länge >0
mit Anfang(p)=Anfang(q) , Ende(p)=Ende(q)

(„Einfache" Masche: Teilwege bilden keine Masche)

Zyklus in gerichteten Graphen:

Einfacher Weg p der Länge >0
mit Ausnahmebedingung Anfang(p)=Ende(p)
d.h. $\forall i \, \forall j \in \{ 0,\ldots,n \}: v_i = v_j \rightarrow i = j \vee \{i,j\} = \{0,n\}$

DAG (directed acyclic Graph): gerichteter Graph ohne Zyklen

Erreichbarkeit, Zusammenhang

v′ ist *erreichbar* von v,
falls ein gerichteter Weg p existiert
mit Anfang(p)=v , Ende(p)=v′ .

Die Relation *„Erreichbarkeit"* ist die *reflexive, transitive Hülle* der Relation E .

G heißt *zusammenhängend*,
wenn zu je zwei Knoten $v,v' \in V$
ein ungerichteter (!) Weg p existiert
mit Anfang(p)=v , Ende(p)=v′

G heißt *stark zusammenhängend*,
wenn zu je zwei Knoten $v,v' \in V$
ein gerichteter (!) Weg p existiert
mit Anfang(p)=v , Ende(p)=v′

wenn jeder Knoten $v' \in V$ von jedem Knoten $v \in V$ erreichbar ist.

Probleme für Graphen

Ist ein Knoten v von einem Knoten v_0 *erreichbar*?

Ist G *(stark) zusammenhängend*?

Ist G *azyklisch?*

Existiert ein *Hamilton-Kreis?*
(Hamilton-Kreis: Zyklus, der ganz V umfaßt).

Existiert ein *Euler-Kreis*?
(Euler-Kreis: Zyklus, der jede Kante genau einmal enthält).

k-Färbungsproblem:
Existiert eine Knotenbeschriftung $\alpha: V \rightarrow \{1,...,k\}$ mit
$\forall v\ \forall v'([v,v'] \in E \rightarrow \alpha(v) \neq \alpha(v'))$

Konstruktion der erreichbaren Knoten

Sei $G=[V,E]$ ein Graph, $v_0 \in V$.
$M(v_0) =_{Def} \{ v \mid v \text{ erreichbar von } v_0 \}$

$M_i(v_0)$ sind die mit Wegen der Länge kleiner gleich i erreichbaren v

Konstruktion:

- Schritt 0: $M_0(v_0) := \{ v_0 \}$
- Schritt i: $M_i(v_0) := M_{i-1}(v_0) \cup \{ v' \mid \exists v \in M_{i-1}(v_0) : [v,v'] \in E \}$
- Abbruch, falls $M_i(v_0) = M_{i-1}(v_0)$

Satz:

Der Algorithmus bricht nach höchstens card(V) Schritten ab. Beim Abbruch gilt $M_{i-1}(v_0) = M(v_0)$.

Konstruktion der erreichbaren Knoten

Folgerungen

Für endliche Graphen G ist entscheidbar,
- ob ein Knoten v' von einem Knoten v erreichbar ist.
- ob G stark zusammenhängend ist.
- ob G zusammenhängend ist.

Wenn v' von einem Knoten v erreichbar ist, so auch mit einem Weg der Länge $l <$ card(V).

Weitere Sätze und Komplexitätsbetrachtungen in Theoretischer Informatik

Baum

Eingangsvalenzen stets 1 oder 0 (Wurzel)

(Gerichteter) Baum: $T = [V,E,r]$
ist ein gerichteter Graph $[V,E]$
mit speziellem Knoten (Wurzel) $r \in V$,
von dem aus alle anderen Knoten
auf genau einem Weg erreichbar sind.

- gerichtet
- zusammenhängend
- ohne Zyklen, ohne Maschen
- Wurzelknoten r eindeutig bestimmt

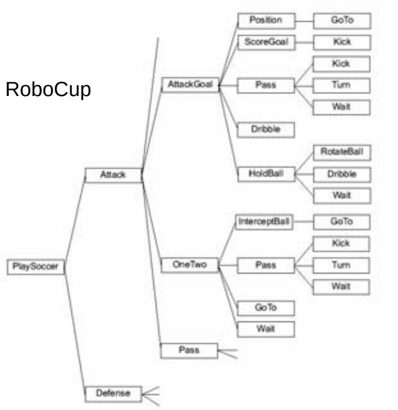
„Ungerichteter Baum“:
jeder Knoten könnte Wurzelknoten sein

Bäume

Hierarchien:

Absichten
und
Unterabsichten

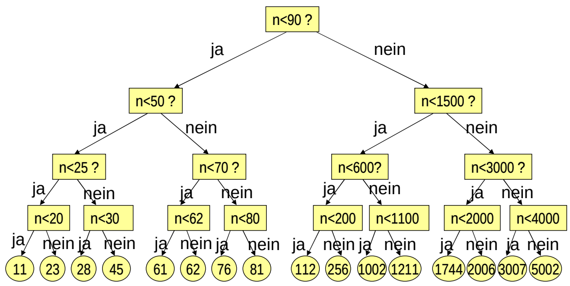
RoboCup



Bäume

Suchbaum

Beschriftungen analog zu Graphen



Baum: Rekursive Definition

Anfang:

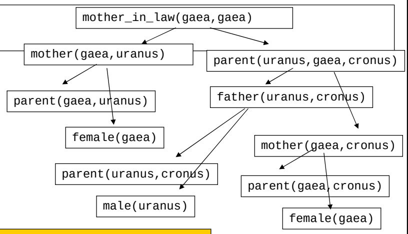
Ein Knoten v ist ein Baum: T=[{v}, ∅, v]

Rekursionsschritt:

Wenn v ein Knoten ist
und wenn T_i = [V_i,E_i, w_i] (i=1,...,n) Bäume sind,
deren Knotenmengen paarweise disjunkt sind,
so ist
$T = [V_1 \cup ... \cup V_n \cup \{v\} , E_1 \cup ... \cup E_n \cup \{ [v, w_i] \mid i = 1,...,n \} , v]$
ein Baum.

Bäume

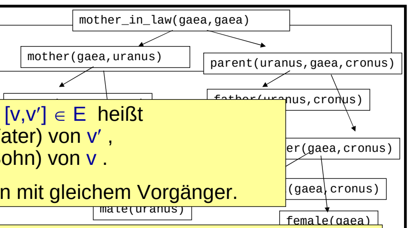
Beweisbaum



Knoten ohne Nachfolger: *Blätter*.
Knoten mit Nachfolger: *innere Knoten*.
Der Knoten ohne Vorgänger: *Wurzel.*

Tiefe: Entfernung von der Wurzel

Bäume



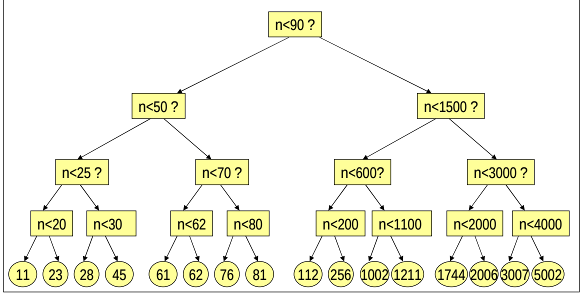
Für Knoten v, v′ mit [v,v′] ∈ E heißt
v der Vorgänger (Vater) von v′ ,
v′ ein Nachfolger (Sohn) von v .

Geschwister: Knoten mit gleichem Vorgänger.

Geordnete Bäume:
Geschwister geordnet (Darstellung: von links nach rechts)

binäre Bäume:
- innere Knoten haben genau 2 Nachfolger
- weiterhin bei Ordnung der Beschriftungen:
  geordnet bzgl. linker/rechter Nachfolger,
  Unterbäume konsistent mit Ordnung)

Binärer Suchbaum



Bäume

Bei Tiefe d und Verzweigungszahl b in den Knoten:

b^d Knoten in jeder Schicht

Baum der Tiefe d hat dann insgesamt
$1 + b + b^2 + ... + b^d$ Knoten

letzte Schicht hat mehr Knoten
als der ganze vorherige Baum (falls b >1)