

Adaptieve Systemen

Adaptieve systemen zijn systemen waarin er een wisselwerking bestaat tussen het systeem en zijn omgeving, zodat beiden steeds transities maken naar veranderlijke toestanden.

Aan de hand van veranderende toestanden van het systeem zal een adaptief systeem zichzelf en/of de omgeving aanpassen om een bepaald doel te bewerkstelligen (simulatie, optimalisatie).

Een lerend adaptief systeem heeft een mogelijkheid om zijn eigen prestatie te meten en kan zijn eigen interne parameters wijzigen om zichzelf te verbeteren (optimaliseren).

Andere geassocieerde benamingen zijn:

- Cybernetica
- Zelf-organiserende systemen
- Complexe adaptieve systemen

Voorbeelden van adaptieve systemen zijn:

- Robots die hun weg kunnen vinden in een bepaalde omgeving
- Lerende systemen die data omzetten in kennis (vb. boomclassificatie)
- Automatisch rijdende auto's of automatische vliegtuig piloten
- Evolutionaire systemen waarin de distributie van de genenpool zich aanpast aan de omgeving
- Economieën waarin goed presterende bedrijven zich expanderen en anderen failliet gaan
- Organismale systemen welke bepaalde essentiële eigenschappen handhaven om te kunnen overleven (zoals homeostase, autonomie)

Vraag: Bedenk een ander voorbeeld van een adaptief systeem

Modelleren m.b.v. agenten

Als een systeem gecontroleerd moet worden, gebruiken we meestal de agent-metafoor.

Een agent krijgt informatie over de toestand van de omgeving d.m.v. inputs, en kiest op basis van zijn interne toestand en de input een actie.

De actie leidt tot een verandering van de toestand van de omgeving.

Verder is een agent autonoom: hij bepaalt zelf zijn actie en kan bepaalde subdoelen tegen elkaar afwegen.

Een agent kan ook communiceren met andere agenten, in dit geval spreken we van een sociale agent in een multi-agent systeem.

De agent kan doelen hebben welke met een beloningsfunctie gemodelleerd kunnen worden.

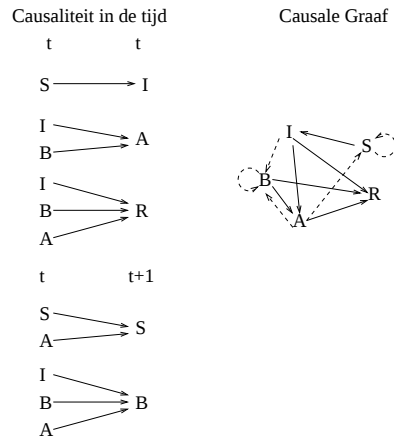
Model voor adaptieve systemen

Een adaptief systeem welke interacteert met een omgeving kan gemodelleerd worden met:

- Een tijd-element $t = \{1, 2, 3, \dots\}$
- De toestand van de omgeving op tijdstip t : $S(t)$
- Input van de agent verkregen op tijdstip t : $I(t)$
- Een interne toestand op tijdstip t : $B(t)$
- Een verzameling mogelijke acties: A , met $A(t)$ de actie op tijdstip t
- Een policy welke de input en interne toestand afbeeldt op een actie van het systeem:
 $\Pi(I(t), B(t)) \rightarrow A(t)$
- Een transitieregel welke de toestand van de omgeving en de actie van het systeem afbeeldt op een volgende toestand van de omgeving:
 $T(S(t), A(t)) \rightarrow S(t+1)$
- Een beloningsfunctie voor het systeem:
 $R(I(t), B(t), A(t)) \rightarrow R(t)$ of
 $R(S(t), A(t)) \rightarrow R(t)$
- Een update functie van de interne toestand van het systeem:
 $U(I(t), B(t), A(t)) \rightarrow B(t+1)$

Causale relaties in het model

We zien bepaalde causale relaties in het model:



Beloningsfunctie

Een agent kan bepaalde doelen hebben welke hij moet bereiken. Hiervoor zou men kwalitatieve Goals kunnen gebruiken.

Een andere mogelijkheid is om een kwantitatief belonings signaal te geven om een actie van de agent te beoordelen.

In de Logica gebruikt men meestal Goals en in decision theory meestal utilities of beloningssignalen.

Het doel van de agent is om zijn som van beloningen verkregen in de toekomst te maximaliseren door het gebruik van een bepaalde policy:

$$\sum_{t=0}^{\infty} \gamma^t R_t$$

Hier is $0 \leq \gamma \leq 1$ de discount factor welke bepaalt in hoeverre lange termijn beloningen mee moeten tellen.

Intelligente agenten

Een intelligente agent kan waarnemen (d.m.v. sensoren), redeneren, voorspellen, en handelen (d.m.v. actuatoren).

Een **rationele agent** handelt om zijn *performance maat* te maximaleren: v.b. doel bereiken met minst mogelijke kosten.

Een **autonome** agent handelt op basis van zijn eigen ervaringen. De agent voert dus niet een vaststaand algoritme uit, maar gebruikt zijn waarnemingen om zijn gedrag te sturen (eventueel ook bij te stellen).

Een agent wordt gestuurd door een **programma** welke draait op een **architectuur** (computer, hardware).

Het programma, de architectuur, en de omgeving bepalen het gedrag van de agent.

Vraag: In hoeverre bepaalt de architectuur het gedrag van een agent?

Totale Systeem Perspectief

Voor het begrijpen van de interactie tussen de agent en de omgeving is het van belang om naar het gehele systeem te kijken.

Een voorbeeld: Bosbrand controle. De entiteiten die een rol spelen zijn bomen, bulldozers, vliegtuigen, vuur, rookkolommen, het weer etc.

Hier zijn bulldozers en vliegtuigen de (controleerbare) autonome agenten.

Soms kan het lastig zijn om te abstraheren van de werkelijkheid: we willen niet alle details opnemen, maar wel een realistische interactie tussen agent en omgeving modelleren.

Vraag: Welke entiteiten spelen een rol in het exploiteren van een restaurant? Wat zijn de agenten?

Een voorbeeld: de bekende kachel

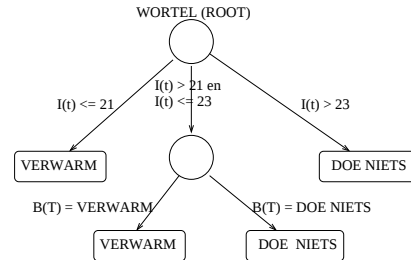
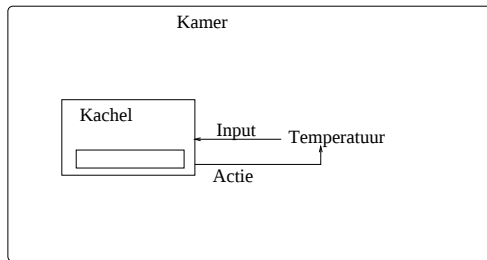
Beschouw een kachel welke ervoor moet zorgen dat de kamertemperatuur gereguleerd wordt.

De kachel heeft een thermometer om de kamertemperatuur op te meten. Dit is de input van het systeem.

De kachel heeft de acties: Verwarm of doe niets.

De temperatuur van de kamer daalt als de kachel uitstaat en wordt hoger als de kachel aanstaat.

Model van de kachel (1)



De gemodelleerde toestand van de omgeving $S(t)$ is de kamertemperatuur.

De input $I(t)$ van de kachel is in dit geval ook de kamertemperatuur op tijdstip t .

Als interne toestand $B(t)$ heeft de kachel enkel als waarden of de kachel reeds aanstond of uitstond (Verwarm of Doe niets).

De acties van de kachel zijn: Verwarm, Doe niets.

Policy van de kachel

Het belangrijkste element is de policy, omdat deze de interactie met de omgeving bepaalt (en eventueel geoptimaliseerd moet worden).

De constructie van een policy kan met de hand gebeuren, maar kan ook geleerd worden.

We kunnen een policy voor de kachel maken, b.v. bestaande uit als-dan regels:

- (1) If $I(t) \leq 21$ Then Verwarm
- (2) If $I(t) > 21$ and $I(t) \leq 23$ and $B(t) ==$ Verwarm Then Verwarm
- (3) If $I(t) > 21$ and $I(t) \leq 23$ and $B(t) ==$ Doe niets Then Doe niets
- (4) If $I(t) > 23$ Then Doe niets

Beslisbomen

Deze policy kunnen we ook met behulp van een beslisboom laten zien:

Model van de kachel (2)

De update functie voor de interne toestand van de kachel is simpel en ziet er als volgt uit:

$$U(*, *, Verwarm) \rightarrow Verwarm$$

$$U(*, *, Doe_niets) \rightarrow Doe_niets$$

Een simpele transitie regel voor de omgeving is als volgt:

$$T(S(t), Verwarm) \rightarrow S(t) + 0.1$$

$$T(S(t), Doe_niets) \rightarrow S(t) - 0.05$$

De beloningsfunctie hebben we alleen nodig voor zelflerende systemen. We zouden deze kunnen instellen als:

$$R(I, *, *) = -|I - 22|$$

Dus het systeem krijgt steeds straf als de temperatuur van de 22 graden afwijkt.

Dynamiek van de wisselwerking

Als we de kachel laten interacteren met de omgeving met een bepaalde begintemperatuur van stel 15 graden, krijgen we een dynamiek van de volgende parameters:

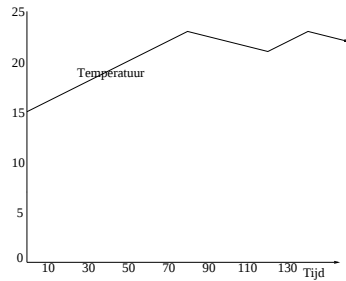
- De toestand van de omgeving
- De input van de kachel (in dit geval gelijk aan de toestand van de omgeving)
- De actie van de kachel
- De verkregen beloning

Voorbeeld: temperatuur

Omgevingen

Er zijn vele verschillende omgevingen, deze kunnen gekarakteriseerd worden aan de hand van de volgende kenmerken:

- **Volledige info/partiële informatie.** Ziet een agent de volledige toestand van een omgeving met zijn/haar sensoren, of slechts een gedeelte hiervan?



	Volledige informatie	Deterministisch	Episodisch	Statisch	Discreet
Omgeving					
Schaken met klok	Yes	Yes	No	Semi	Yes
Schaken zonder klok	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi rijden	No	No	No	No	No
Medische diagnose	No	No	No	No	No
Beeld analyse	Yes	Yes	Yes	Semi	No
Interactieve engelse leraar	No	No	No	No	Yes

- **Deterministisch vs. Niet-deterministisch.** Wordt de volgende toestand van de omgeving uniek bepaald door de huidige toestand en de gekozen actie van de agent, of is er een kansverdeling naar opvolgende toestanden?
- **Episodisch vs. Niet episodisch.** Wordt een agent steeds voor 1 losstaande keuze gesteld welke onafhankelijk is van volgende acties, of zit de agent in een wereld waarbij de hele sequentie van acties een rol speelt.
- **Statisch vs. Dynamisch.** Verandert de wereld tijdens het handelsproces, of niet. Als de overgangen tussen toestanden veranderen, dan is de omgeving dynamisch en speelt tijd een rol. Als de prestatie maat (belonings functie) verandert heet de omgeving **semi-dynamisch**.
- **Discreet vs. Continu.** Is de toestand van de wereld discreet gerepresenteerd (vb. schaken) of niet, v.b. robot in continue omgeving met snelheid, x en y positie etc.

Partiële informatie, niet-deterministische, niet episodische, dynamische, continue omgevingen zijn het moeilijkst voor de optimalisatie van agenten.

Simulatie kan altijd, maar een goed voorspellend model is soms weer heel moeilijk te maken (vb. weersvoorspelling).

De complexiteit van het gedrag van een agent hangt vaak af van de complexiteit van de omgeving (vb. Simon's Ant).

Vraag: bedenkt zelf een omgeving en bepaal de karakteristieken van deze omgeving.

De interne toestand

Dikwijls wordt er geen interne toestand gebruikt, deze **reactieve** agent mapt direct input op actie.

In sommige gevallen krijgt de agent dezelfde input in verschillende states. Zijn optimale acties kunnen dan niet van elkaar afwijken. In zulke gevallen, gebruiken we de interne toestand om de huidige input te disambigueren.

De interne toestand vat het verleden samen, alle vorige inputs en acties kunnen de huidige interne toestand bepalen.

Als de actie veel van de interne toestand en weinig van de input afhangt, kunnen we spreken van een "introverte" agent.

In het geval van psychoses, gaat de interne toestand zijn eigen weg en kan het geloof erin door nieuwe inputs en acties moeilijk weerlegd worden.

Vraag: Hoe verandert de yogi zijn interne toestand met meditatie?

Multi-agent systemen

Als er meerdere agenten zijn, spreken we van een multi-agent systeem (MAS).

Hoewel het hele systeem in veel gevallen ook met 1 superagent gemodelleerd kan worden, heeft het opsplitsen van het systeem in meerdere gedecentraliseerde agenten bepaalde voordelen:

- Robuustheid
- Snelheid (Distributed computing)

Voorbeelden van omgevingen

- Eenvoudigere uitbreidbaarheid of verandering
- Information hiding (privacy)

Model van een multi-agent systeem

Als we te maken hebben met een multi-agent systeem, kunnen we de enkele agenten modelleren als hiervoor (met inputs, acties, interne toestand, policy, beloningsfunctie, update functie)

In sommige gevallen wordt het systeem ook uitgebreid met communicatie tussen agenten.

De agenten hebben dan de beschikking over communicatie signalen (een taal) en mappen inputs en interne toestand op een communicatie signaal.

Zo'n communicatie signaal kan aan alle agenten worden gestuurd (broadcasting), maar ook naar een enkele agent.

Communicatie is van belang als de agenten moeten coopereren. De agenten moeten dan op een bepaalde manier gecoördineerd worden.

Complexe Adaptieve Systemen

Bepaalde systemen bestaande uit meerdere entiteiten worden in sommige gevallen complexe adaptieve systemen genoemd.

Het verschil met multi-agent systemen is dat het niet zozeer gaat om de controle en optimalisatie van een systeem, maar meer om simulatie. Entiteiten worden hier vaak als niet rationeel verondersteld.

In complexe adaptieve systemen kunnen simpele regels een complex gedrag genereren als er meerdere entiteiten met elkaar kunnen interacteren.

We zeggen dan dat de totale dynamiek **emergeert** uit de wisselwerking tussen de entiteiten en de omgeving.

Vervolg Complexe adaptieve systemen

Voorbeelden van processen die we met complexe adaptieve systemen kunnen modelleren zijn:

- Het verkeer bestaande uit vele weggebruikers

- Bosbranden bestaande uit vele bomen die de brand doorpropageren
- Infectie ziekten bestaande uit virussen en virusdragers
- Magnetisme bestaande uit deeltjes welke positief of negatief geladen kunnen zijn
- Ecologieën bestaande uit vele soorten dieren welke elkaar opeten en nakomelingen produceren.
- Economische markten welke bestaan uit vele verschillende beleggers

Vraag: welke van deze processen kunnen beter met MAS gemodelleerd worden?

Predator-prooi systeem

Een simpel voorbeeld van een systeem bestaande uit meerdere entiteiten is een predator-prey systeem.

De predator gaat op zoek naar prey (prooi) om op te eten en plant zich voort.

De prooi gaat zelf ook op zoek naar voedsel, probeert de predator te ontwijken, en plant zich voort.

De populatie van prooien en predatoren hangen van elkaar af.

Als er veel predatoren zijn, daalt de populatie prooien.

Als er dan weinig prooien meer zijn, daalt de populatie predatoren.

Als er weinig predatoren meer zijn, groeit de populatie prooien.

Lotka-Volterra vergelijkingen

We kunnen zo'n systeem vatten m.b.v. bepaalde regels. Een bekend model is het gebruik van Lotka-Volterra vergelijkingen.

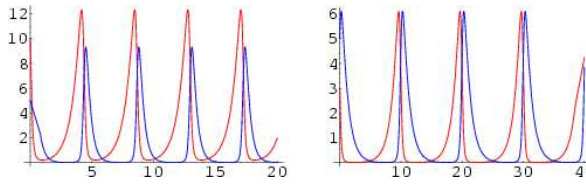
We noemen de grootte van de populatie prooien x en de de grootte van de populatie predatoren y .

Nu stellen we als toestand $S(t) = (x(t), y(t))$ met:

$$x(t+1) = x(t) + Ax(t) - Bx(t)y(t)$$

$$y(t+1) = y(t) - Cy(t) + Dx(t)y(t)$$

Voorbeelden dynamiek:



Vraag: Teken de dynamiek op de x - en y -assen

Stabiel punt

De dynamiek kan leiden tot een stabiel punt, periodisch zichzelf herhalen, of chaotisch zijn (altijd veranderlijk)

Voor een stabiel punt is het van belang dat alles gelijk blijft. Dus $x(t+1) = x(t)$ en $y(t+1) = y(t)$

We kunnen een stabiel punt $(x(*), y(*))$ vinden door de vergelijkingen:

$$x(*) = x(*) + Ax(*) - Bx(*)y(*) \quad (1)$$

$$0 = A - By(*) \quad (2)$$

$$y(*) = \frac{A}{B} \quad (3)$$

$$y(*) = y(*) - Cy(*) + Dx(*)y(*) \quad (4)$$

$$0 = -C + Dx(*) \quad (5)$$

$$x(*) = \frac{C}{D} \quad (6)$$

Periodieke cyclen

In bepaalde gevallen treedt er een herhaling op in een systeem. We spreken dan van een periodieke cykel.

We hebben te maken met een periodieke cykel van lengte n als:

$$\begin{aligned} S(t) &= S(t+n) \\ S(t+1) &= S(t+n+1) \\ &\vdots \\ S(t+n-1) &= S(t+2n-1) \end{aligned}$$

Een systeem is chaotisch als het niet in een stabiel punt terecht komt, en er ook geen periodieke cyclen zijn. Het systeem herhaalt zich dan nooit.

Het is vaak moeilijk te detecteren of een systeem echt chaotisch is, de cykel zou erg lang kunnen zijn!

Vraag: denk je dat chaos in een computerprogramma mogelijk is?

Inhoud van dit vak

We gaan de volgende onderwerpen behandelen in de vervolg colleges:

- Cellulaire Automaten
- Artificial Life
- Biologische Adaptieve Systemen
- Co-evolutie
- Robotica
- Evolutionary Computation (2 colleges)
- Lerende Machines
 1. Decision Trees
 2. Bayesian Learning
 3. Neurale Netwerken
 4. Self-organizing maps
- Reinforcement leren (3 colleges)

Praktische Informatie

De toetsing van het vak bestaat uit:

- 2 Deeltentamens (beide 40% van het eindcijfer)
- Inlever opgaven (20% van het eindcijfer)
- Practicum opdrachten (Verplichte voldoende)

De inlever opgaven dienen binnen een week nadat de opgaven verspreid zijn, ingeleverd te worden.

Een aantal van de practicum opgaven zijn op te halen van het lokweb (www.ou.nl/lokweb).

Conclusie

We hebben inzicht opgedaan in adaptieve systemen welke interacteren met een omgeving.

Adaptieve systemen kunnen goed gemodelleerd worden d.m.v. agenten met policies en een interne toestand.

Als er meerdere agenten zijn, spreken we van multi-agent systemen (MAS). De agenten in een MAS kunnen dikwijls met elkaar communiceren.

Veel natuurlijke processen kunnen gemodelleerd worden met multi-agent systemen (optimalisatie) of complexe adaptieve systemen (simulatie).

De dynamiek van een systeem resulteert uit de begintoestand en de regels welke de nieuwe toestand bepalen.

De dynamiek kan leiden naar een stabiel punt, kan periodiek gedrag vertonen of chaotisch zijn.