

Tentamen IN2205/IN2420 Kennissystemen

27 Juni 2007
14:00–17:00

Dit tentamen heeft als doel je kennis en begrip van Michael Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems* te testen. Het bevat 18 vragen. In totaal zijn 60 punten te behalen. Het cijfer wordt zó berekend dat als je de multiple-choicevragen slechts willekeurig invult het verwachte cijfer 1 is.

Het is niet toegestaan het boek te gebruiken tijdens het tentamen. Schrijf voor de **multiple-choice**vragen de letter met het beste antwoord op een apart blad papier. Je mag op de **open** vragen antwoorden in het Nederlands of het Engels. Motiveer je antwoorden, maar geef geen irrelevante informatie in je antwoorden; hiervoor worden namelijk punten afgetrokken.

Controleer voor het inleveren of op elk antwoordblad je naam, studienummer en de juiste vakcode (IN2205 of IN2420) staan vermeld.

Succes!

Multiple-choicevragen

Opgave 1

1 punt

Welke uitspraak omtrent expert systemen is *niet* juist?

- A. Expert systemen scheiden kennis en redeneren.
- B. Expert systemen maken expert kennis toegankelijk.
- C. Expert systemen leren automatisch nieuwe regels.
- D. Expert systemen kunnen een verklaring geven voor de gevonden oplossing.

Antwoord: C.

Expert systemen leren automatisch nieuwe regels.

Opgave 2

1 punt

Waarvoor gebruiken expert systemen conflict resolution?

- A. Om te bepalen of forward chaining of backward chaining gebruikt moet worden.
- B. Om te bepalen welke consequent de grootste kans heeft.
- C. Om te bepalen welke regel mag vuren als meer dan één regel kan vuren.
- D. Om te bepalen welke regels tot een bepaalde conclusie hebben geleid.

Antwoord: C.

Om te bepalen welke regel mag vuren als meer dan een regel kan vuren.

Opgave 3

1 punt

Problemen samenhangend met het onttrekken van kennis aan een menselijke expert teneinde deze kennis in computers te modelleren worden wel samengevat met de term:

- A. Knowledge generation bottleneck.
- B. Knowledge production bottleneck.
- C. Knowledge acquisition bottleneck.
- D. Knowledge execution bottleneck.

Antwoord: C.

Knowledge acquisition bottleneck (blz. 9).

Opgave 4

1 punt

Wat is een gebruikelijke manier om de doorsnede (AND) van twee fuzzy sets te definiëren?

- A. $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$.
- B. $\mu_{A \cap B}(x) = \max[\mu_A(x), \mu_B(x)]$.
- C. $\mu_{A \cap B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x)$.
- D. Geen van bovenstaande antwoorden.

Antwoord: A.

De doorsnede wordt meestal gedefinieerd als $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$. Een alternatief is $\mu_{A \cap B}(x) = \mu_A(x) \times \mu_B(x)$.

Opgave 5

1 punt

Stel we hebben de fuzzy sets *koud* met $\mu_{koud}(x) = (1/0, 0.8/5, 0.6/10, 0.4/15, 0.2/20, 0/25)$ en *warm* met $\mu_{warm}(x) = (0/0, 0.1/5, 0.2/10, 0.4/15, 0.7/20, 1/25)$ wat is de membershipfunctie van de fuzzy set *niet(koud of warm)* als we gebruik maken van de standaardoperatoren?

- A. $\mu(x) = (0/0, 0.2/5, 0.4/10, 0.6/15, 0.8/20, 1/25)$.
- B. $\mu(x) = (0/0, 0.2/5, 0.4/10, 0.6/15, 0.3/20, 0/25)$.
- C. $\mu(x) = (1/0, 0.9/5, 0.8/10, 0.6/15, 0.8/20, 1/25)$.
- D. $\mu(x) = (0/0, 0.1/5, 0.2/10, 0.4/15, 0.2/20, 0/25)$.

Antwoord: B.

$$\begin{aligned} \mu_{\text{koud of warm}} &= \mu_{\text{koud}} \cup \mu_{\text{warm}} \\ &= \max(\mu_{\text{koud}}, \mu_{\text{warm}}) = (1/0, 0.8/5, 0.6/10, 0.4/15, 0.7/20, 1/25) \\ \mu_{\text{niet(koud of warm)}} &= 1 - \mu_{\text{koud of warm}} = (0/0, 0.2/5, 0.4/10, 0.6/15, 0.3/20, 0/25). \end{aligned}$$

Opgave 6

1 punt

In een fuzzy systeem voor wachttijden worden de membershipfuncties $\mu_{kort}(x)$ en $\mu_{lang}(x)$ gebruikt. Een gemiddelde wachttijd wordt aangeduid als *niet lang en niet kort*. Welke van onderstaande membershipfuncties geeft dit correct weer.

- A. $\mu_{kort} + \mu_{lang} - (\mu_{kort} \times \mu_{lang})$.
- B. $\max(1 - \mu_{kort}, 1 - \mu_{lang})$.
- C. $1 - \min(\mu_{kort}, \mu_{lang})$.
- D. $1 - \max(\mu_{kort}, \mu_{lang})$.

Antwoord: D.

Gebruik de wetten van De Morgan (blz. 102):

$$\neg\mu_{kort} \cap \neg\mu_{lang} = \neg(\mu_{kort} \cup \mu_{lang}) = 1 - \max(\mu_{kort}, \mu_{lang}).$$

Opgave 7

1 punt

Tussen twee classes *A* en *B* in een frame-based systeem bestaat een *a-part-of* relatie. Er is hier sprake van een ...

- A. Generalisatie.
- B. Associatie.
- C. Aggregatie.
- D. Overerving.

Antwoord: C.

Aggregatie.

Opgave 8

1 punt

Welke soort learning is bij het Hopfield netwerk aan de orde?

- A. Unsupervised learning.
- B. Supervised learning.
- C. Reinforcement learning.
- D. Hebb learning.

Antwoord: B.
Supervised learning.

Opgave 9

1 punt

Een Bidirectional associative memory (BAM) is een ...

- A. recurrent netwerk.
- B. multilayer netwerk met backpropagation.
- C. feedforward netwerk.
- D. self-organising netwerk.

Antwoord: A.
recurrent netwerk (blz. 196).

Opgave 10

1 punt

Welke uitspraak is *niet* waar met betrekking tot feedforward multilayer neurale netwerken?

- A. Neurale netwerken kunnen met binaire en continue data overweg.
- B. Neurale netwerken zijn ongevoelig voor ruis in de data.
- C. Neurale netwerken zijn ongevoelig voor overtrainen.
- D. Neurale netwerken zijn gevoelig voor lokale maxima.

Antwoord: C.
Neurale netwerken zijn ongevoelig voor overtrainen.

Opgave 11

1 punt

Waarvoor staat de constante α in de perceptron learning rule $w_i = w_i(p) + \alpha \times x_i(p) \times e(p)$?

- A. De threshold, een positieve constante groter dan 1.
- B. De learning rate, een positieve constante kleiner dan 1.
- C. Het momentum, een positieve constante kleiner dan 1.
- D. De forgetting factor, een positieve constante kleiner dan 1.

Antwoord: B.
De learning rate, een positieve constante kleiner dan 1.

Opgave 12

1 punt

Waarvoor dient de mutatie-operator in een genetisch algoritme?

- A. Het bepaalt welke stukken chromosoom van de ene en welke van de andere ouder worden overgenomen.
- B. Het voorkomt dat het zoekalgoritme in een lokaal optimum blijft steken.
- C. Het vult de chromosomen aan tot de juiste lengte.
- D. Het zorgt dat chromosomen met een hoge fitness een grotere kans hebben om te overleven.

Antwoord: B.

Het voorkomt dat het zoekalgoritme in een lokaal optimum blijft steken.

Opgave 13

1 punt

We willen een systeem ontwikkelen dat een diagnose kan stellen bij motorproblemen in een auto. Welke techniek kunnen we hier het beste voor inzetten?

- A. Een fuzzy expert systeem.
- B. Een recurrent neuraal netwerk.
- C. Een self organising neuraal netwerk.
- D. Een genetisch algoritme.

Antwoord: A.

Een fuzzy expert systeem. Voor dit soort problemen kunnen we regels opstellen die ook door experts gebruikt worden. Het gedrag van veel componenten in een motor zal echter op een continue schaal uitgedrukt kunnen worden (oliepijl, warmte) waarbij het niet aankomt op een precieze waarde, maar eerder begrippen als 'het oliepijl is laag', 'de motor wordt erg warm' gebruikt zullen worden. Een implementatie met fuzzy regels ligt dus voor de hand.

Open vragen

Opgave 14

9 punt

Stel, we hebben een knowledge base met de volgende regels:

1. IF c AND k THEN l
2. IF d THEN e
3. IF h AND j THEN m
4. IF a AND c THEN d
5. IF l AND g THEN f
6. IF e AND f AND j THEN b
7. IF e AND g THEN h
8. IF m AND k THEN l

In de database komen de volgende feiten voor: a, c, g, j . De volgorde van de regels in de knowledge base is bepalend in geval van conflicten.

- (a) (3 punten) Laat zien hoe de inference engine aantooit of doel b waar is. Geef aan welke feiten op de stack worden geplaatst en hoe de database verandert.

Antwoord:

Database: a, c, g, j .

Doelen: b .

Stack: 6. Doelen e, f, b . (j staat al in de database)

(We proberen doel e te vervullen. We zouden ook eerst doel f kunnen bekijken).

Stack: 2, 6. Doelen: d, e, f, b .

Stack: 4, 2, 6. Doelen: d, e, f, b .

Regel 4 vuurt, database: a, c, d, g, j .

Stack: 2, 6. Doelen e, f, b .

Regel 2 vuurt, database: a, c, d, e, g, j .

Stack: 6. Doelen: f, b .

Stack: 5, 6. Doelen: l, f, b . (g staat al in de database)

Doel l kan via regel 1 en via regel 8 vervuld worden. In geval van conflicten is de volgorde van de regels bepalend, we kijken dus eerst naar regel 1.

Stack: 1, 5, 6. Doelen: k, l, f, b .

Er is geen regel die k kan vervullen, regel 1 kan niet vuren. We proberen l te vervullen via regel 8.

Stack: 8, 5, 6. Doelen: m, k, l, f, b .

We moeten wederom k vervullen en kunnen dus hier stoppen. Als we eerst hadden geprobeerd m te vervullen:

Stack: 3, 8, 5, 6. Doelen: h, m, k, l, f, b .

Stack: 7, 3, 8, 5, 6. Doelen: e, g, h, m, k, l, f, b .

Regel 7 vuurt, database: a, c, d, e, h, g, j .

Stack: 3, 8, 5, 6. Doelen: m, k, l, f, b .

Regel 3 vuurt, database: a, c, d, e, h, g, j, **m**.
Stack: 8, 5, 6. Doelen: k, l, f, b.

Er zijn geen doelen meer die vervuld kunnen worden.
Doel *b* is niet waar.

- (b) (3 punten) Laat zien hoe de inference engine alle feiten afleidt die kunnen worden afgeleid. Geef per *cycle* aan welke regels vuren en hoe de database verandert.

Antwoord:

Regel vuurt	database
4	a, c, d , g, j.
2	a, c, d, e, g, j.
7	a, c, d, e, g, h , j.
3	a, c, d, e, g, h, j, m .

Meer regels kunnen niet vuren.

- (c) (3 punten) Stel we willen een systeem voor handschriftherkenning ontwikkelen. Zou het handig zijn hier een expertsysteem voor te gebruiken? Waarom wel/niet?

Antwoord: Nee, dit zou niet handig zijn. Letters hebben weliswaar specifieke vormen, maar ieder handschrift is anders. Ook binnen een handschrift kan variatie optreden. We kunnen hier beter voor een neurale netwerk kiezen dat handschriften kan leren van voorbeelden en in staat is te generaliseren en daardoor afwijkingen op te vangen.

Opgave 15

4 punten

We hebben twee hypothesen H_1 en H_2 en observeren achtereenvolgens bewijs E_1 , E_2 en E_3 . De a priori kans van hypothese H_1 is 0.6 en die van H_2 is 0.3. Tabel 1 geeft de conditionele kansen van het bewijs gegeven de hypothesen:

	E_1	E_2	E_3
H_1	0.5	0.3	0.1
H_2	0.4	0.7	0.2

Wat is de kans (afgerond op drie cijfers achter de komma) dat hypothese H_1 juist is gegeven het bewijs E_1 , E_2 en E_3 als we gebruik maken van Bayesiaans redeneren?

Antwoord:

$$p(H_1|E_1, E_2, E_3) = \alpha P(H_1)P(E_1|H_1)P(E_2|H_1)P(E_3|H_1)$$

$$p(H_2|E_1, E_2, E_3) = \alpha P(H_2)P(E_1|H_2)P(E_2|H_2)P(E_3|H_2)$$

$$p(H_1|E_1, E_2, E_3) = \alpha 0.6 \times 0.5 \times 0.3 \times 0.1 = \alpha 0.009$$

$$p(H_2|E_1, E_2, E_3) = \alpha 0.3 \times 0.4 \times 0.7 \times 0.2 = \alpha 0.0168$$

$$\alpha = 1/(0.009 + 0.0168) = 1/0.258$$

$$p(H_1|E_1, E_2, E_3) = 0.349$$

$$p(H_2|E_1, E_2, E_3) = 0.651$$

Opgave 16

10 punten

- (a) (1 punt) Wat is een fuzzy set?

Antwoord: Een verzameling waarvan elementen in zekere mate lid kunnen zijn (In tegenstelling tot gewone verzameling waarbij een element in de verzameling zit of niet). Een lengte van 1.85 m kan bijvoorbeeld 0.1 lid zijn van de fuzzy set *klein* en 0.8 lid van de fuzzy set *groot*. Let op: elementen in een fuzzy set hebben een lidmaatschapswaarde, dit heeft niets te maken met kansen, zoals sommige geantwoord hebben.

- (b) (1 punt) Wat zijn *hedges* die gebruikt worden in combinatie met fuzzy sets?

Antwoord: hedges zijn functies waarmee fuzzy sets kunnen worden versterkt of afgezwakt. In termen van linguïstische variabelen komen zij overeen met begrippen als ‘erg’ ($\mu(x)^2$) en ‘min of meer’ ($\sqrt{\mu(x)}$).

- (c) (5 punten) Laat met behulp van grafieken zien hoe *Sugeno* inferentie werkt. Behandel achtereenvolgens: Fuzzificatie, Regel evaluatie, Aggregatie en Defuzzificatie.

Antwoord: Bij deze opgave wil ik plaatjes als op bladzijde 113 van het boek zien. Uit deze grafieken moet duidelijk worden wat er gebeurt, bijvoorbeeld welke functies gebruikt worden voor regel evaluatie.

- (d) (3 punten) Wat is het verschil tussen Mamdani en Sugeno inferentie?

Antwoord: Bij Mamdani inferentie is de consequent van een fuzzy regel een fuzzy set. Bij Sugeno is het een functie of een constante. Dit maakt met name de defuzzificatie stap eenvoudiger en sneller. In plaats van een center of gravity hoeft slechts een gewogen gemiddelde berekend te worden.

Opgave 17

12 punten

- (a) (4 punten) Waarin verschillen een perceptron en een multilayer neural network? Noem in je antwoord de topologie en de werking van de netwerken en het soort taken dat met deze netwerken opgelost kan worden.

Antwoord: Een perceptron bestaat uit een enkel neuron. Bij iedere input hoort een gewicht. Het neuron berekent met deze gewichten een gewogen som van de inputs. Als deze som hoger is dan een treshold dan vuurt het neuron. Een step-functie wordt gebruikt als activatie functie.

Een multilayer neurale netwerk bestaat uit meerdere neuronen, geordend in lagen: Een input laag, een of meer verborgen (hidden) lagen en een output laag. De inputs van een neuronen in de hidden en output lagen zijn neuronen in de voorgaande laag. Als activatiefunctie wordt meestal een sigmoid functie gebruikt. Voor training van het netwerk wordt het backpropagation algoritme gebruikt.

Een perceptron kan slechts lineair scheidbare problemen oplossen (zoals de logische AND en OR functies of classificatie van elementen in duidelijk gescheiden groepen). Een multilayer netwerk kan in theorie elke functie benaderen. Mogelijke toepassingen zijn: classificatie van elementen waarvan de verzamelingen niet lineair te scheiden zijn, handschriftherkenning of numerbordherkenning.

- (b) (3 punten) Beschrijf (in woorden) het back-propagation trainingsalgoritme voor multilayer neurale netwerken.

Antwoord: Voor het backpropagation algoritme moeten voorbeelden beschikbaar zijn waarvan de gewenste output bekend is. Elk voorbeeld wordt in het netwerk ingevoerd. De door het netwerk gegeven uitkomst wordt vergeleken met de gewenste output. Net als bij het perceptron wordt het verschil gebruikt om de gewichten van het netwerk bij te stellen. Dit

wordt gedaan door error gradienten terug te propageren door het netwerk. Deze cyclus wordt herhaald totdat het netwerk een voldoende kleine fout maakt (bij voorkeur op een aparte testset, zodat we kunnen voorkomen dat het netwerk de training data leert, maar geen andere gevallen meer kan herkennen).

- (c) (2 punten) Leg uit hoe het leren in een neuraal netwerk kan worden versneld door het introduceren van een *momentum term*.

Antwoord: Een momentum term neemt de vorige wijziging van een gewicht mee in de nieuw wijziging van het gewicht:

$$\Delta w_{jk}(p) = \beta \times \Delta w_{jk}(p-1) + \alpha \times y_j(p) \times \delta_k(p),$$

waarbij $\beta \in [0, 1)$ de momentum constante is.

Dit komt de stabiliteit van het trainingsproces ten goede. De stappen die genomen worden zijn groter in eenvoudige zoekruimtes en kleiner in zoekruimtes met veel pieken en dalen.

- (d) (3 punten) Een Kohonen netwerk maakt gebruik van competitive learning. Hoe werkt competitive learning?

Antwoord: Bij competitive learning mag slechts 1 neuron (of enkele neuronen) zijn gewichten aanpassen in een trainingscyclus. Dit wordt bijvoorbeeld gebruikt bij Kohonen netwerken, waarbij het neuron waarvan de gewichtenvector het dichtst bij de inputvector (euclidische afstand) wint. Op deze manier leert het netwerk de verdeling van de inputs over de inputruimte.

Opgave 18

12 punten

Stel we hebben een groot aantal metingen die voldoen aan een ons onbekende functie $f(a, b)$. Tabel 2 geeft enkele waarden. We willen de functie bepalen met behulp van *genetisch programmeren* en

Tabel 2:

a	b	$f(a, b)$
1	1	4
2	3	15.14626437
3	3	21.46410162
4	3	27.73205081
5	5	54.47213595
10	10	206.3245553

nemen daarbij aan dat het verband tussen input en output uit te drukken valt met behulp van de volgende primitieve functies: $+$, $-$, $*$, $/$ en $\sqrt{\quad}$. Beschrijf hoe dit in zijn werk gaat. Behandel de volgende onderwerpen:

- (a) (1 punt) De beginpopulatie.

Antwoord: We maken een beginpopulatie van N willekeurig gegenereerde programma's (bijvoorbeeld LISP S-expressies) bestaande uit combinaties van de primitieve functies en de variabelen a en/of b .

Let op: de vraag gaat over *genetisch programmeren*, niet over genetische algoritmen.

- (b) (2 punten) De fitness functie.

Antwoord: Een fitnessfunctie moet aangeven hoe goed een gegenereerd programma is. De functie zelf is onbekend (dat is nu juist waar we naar op zoek zijn). Maar we kunnen ieder

programma een output laten genereren aan hand van de combinaties van a en b in de tabel en deze vergelijken met de gewenste output. Om een fitnesswaarde te verkrijgen kunnen we bijvoorbeeld de som over alle verschillen nemen. Hoe dichter dit getal bij 0 komt hoe beter.

- (c) (1 punt) Keuze van een genetische operator.

Antwoord: Bij genetisch programmeren wordt met een bepaalde kans gekozen voor cross-over, mutatie of cloning. De som van deze drie kansen is 1.

- (d) (2 punten) Selectie van een of meerder programma's met behulp van 'roulette wheel' selectie.

Antwoord: De oppervlakte behorende bij een programma op het roulette rad komt overeen met de fitness ratio. Dit is de fitness van het programma gedeeld door de som van de fitnesswaarden van alle programma's. Als cross-over is gekozen in de vorige stap, worden twee programma's geselecteerd door 'aan het roulette wiel te draaien', bij mutatie en cloning wordt één programma geselecteerd.

- (e) (2 punten) Cross-over.

Antwoord: Bij cross-over worden subexpressies van programma's verwisseld. Als we de programma's voorstellen als bomen, dan worden takken van deze bomen verwisseld.

- (f) (2 punten) Mutatie.

Antwoord: Bij mutatie wordt of een variable in een programma veranderd in een andere variabele of een primitieve functie in een andere primitieve functie (met evenveel parameters). Het resultaat is dus altijd een correct programma.

- (g) (1 punt) Cloning.

Antwoord: In geval van cloning wordt een programma simpelweg gekopieerd naar de nieuwe populatie.

- (h) (1 punt) Nieuwe populatie.

Antwoord: Stappen (c), (d), (e) en (f) worden doorlopen totdat de omvang van de nieuwe populatie N is. De nieuwe populatie vervangt dan de oude populatie in de volgende iteratie. Dit gaat een van te voren bepaald aantal iteraties door of totdat weinig verandering meer optreedt.

Einde van het tentamen

Controleer voor de zekerheid of je alle vragen hebt beantwoord. Het zouden er 18 moeten zijn.