

Tentamen IN2205 Kennissystemen

21 Januari 2010, 14:00–17:00

- Dit tentamen heeft 5 meerkeuzevragen in totaal goed voor 10 punten en 7 open vragen met in totaal 40 punten.
- Wat betreft de meerkeuzevragen:
 - Er is voor iedere vraag telkens maar één goed antwoord mogelijk.
- Wat betreft de open vragen:
 - Geef antwoord in correct Nederlands of Engels en schrijf leesbaar (gebruik eerst kladpapier).
 - Motiveer je antwoorden.
 - Geef geen irrelevante informatie. Dit kan leiden tot puntenaftrek.

Vraag:	6	7	8	9	10	11	12	Totaal:
Punten:	6	4	7	2	10	8	3	40

- Het gebruik van boek of aantekeningen tijdens dit tentamen is niet toegestaan.
- Het gebruik van een rekenmachine is toegestaan.
- Voordat je je antwoorden inlevert, controleer of op ieder blaadje je naam en studienummer staat en geef het aantal ingeleverde bladen aan op (tenminste) de eerste pagina.
- De tentamenstof bestaat uit hoofdstukken 1,2,3,4,6,7,8 en 9 uit *Artificial Intelligence* van Michael Negnevitsky, behalve secties 3.4, 3.5, 8.2, 9.1 en 9.7, het artikel *Swarm Smarts* van Eric Bonabeau *et. al.* en de lecture notes over Bayesian Reasoning en Swarm Intelligence.
- Uiteraard komen in één tentamen niet alle onderwerpen aan bod. Trek daarom op basis van dit tentamen geen conclusies over stof die nooit getoetst wordt.
- Totaal aantal pagina's: 6.

Succes!

Multiple-choicevragen

1. (2 punten) Bij een fuzzy expertsysteem dat gebruik maakt van Sugeno inferentie is de consequent van een fuzzy regel ...
 - A. een fuzzy set.
 - B. een fuzzy waarde.
 - C. een functie van de inputs.**
 - D. het zwaartepunt van de fuzzy sets in de antecedent.
2. (2 punten) Als een element x met lidmaatschapswaarde $\mu_a(x)$ in een fuzzy set A en met lidmaatschapswaarde $\mu_b(x)$ in een fuzzy set B voorkomt, dan moet gelden:
 - A. $\mu_a(x) + \mu_b(x) \leq 1.0$
 - B. $\mu_a(x) + \mu_b(x) = 1.0$
 - C. $\mu_a(x) + \mu_b(x) \geq 1.0$
 - D. $\mu_a(x) + \mu_b(x) \leq 2.0$**
3. (2 punten) Het backpropagation algoritme is een vorm van ...
 - A. unsupervised learning.
 - B. reinforcement learning.
 - C. supervised learning.**
 - D. Geen van bovenstaande antwoorden.
4. (2 punten) Bij evolution strategies ...
 - A. komt mutatie neer op een bit flip.
 - B. ligt de kans op een mutatie meestal tussen 0.001 en 0.01.
 - C. veranderen alle parameters van een oplossing tegelijk.**
 - D. wordt geen fitnessfunctie gebruikt.
5. (2 punten) De update-regel voor Particle Swarm Optimization (PSO) luidt:

$$v_{ij}(t+1) = \phi v_{ij}(t) + c_1 r_{1j} [y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j} [\hat{y}_{ij}(t) - x_{ij}(t)] \quad (1)$$

Hoe kunnen we er het best voor zorgen dat de particles de oplossingsruimte uitgebreider doorzoeken?

- A. ϕ verhogen.**
- B. c_1 verhogen.
- C. c_2 verhogen.

Open vragen

6. Geef voor ieder van de onderstaande vraagstukken aan welke van de in dit vak behandelde oplossingsmethoden je zou gebruiken, motiveer je antwoorden.
 - (a) (2 punten) Een software agent in een computerspel moet in een doolhof het poppetje van de speler te pakken zien te krijgen. We willen de software agent zelf het juiste gedrag laten leren. Ga er bij je antwoord vanuit dat de agent alleen horizontaal en verticaal kan bewegen en in deze richtingen telkens 3 stappen vooruit kan kijken, mits er niet eerder al een muur staat.
 - (b) (2 punten) Een tuinsproeier, aan hand van de luchtvochtigheid, droogte van de grond e.d. zelf laten bepalen wanneer en hoe lang te sproeien.
 - (c) (2 punten) Oplossen van een Sudoku.

7. (4 punten) Stel, we hebben een knowledge base met de volgende regels:

1. IF q AND a THEN r
2. IF t AND p THEN y
3. IF b THEN s
4. IF c AND v THEN l
5. IF x OR y THEN z
6. IF d THEN q
7. IF r THEN x
8. IF c THEN t
9. IF t AND s THEN p
10. IF a THEN k
11. IF k THEN v

In de database komen de volgende feiten voor: **a, b, c**.

We gaan uit van een seriële (depth-first) verwerking van regels. Laat zien hoe de inference engine aantooft of doel **z** waar is. Geef aan welke feiten op de stack worden geplaatst en hoe de database verandert.

Antwoord:			
stack	doelen	database	
	z	a, b, c	
5	x, z	a, b, c	
5,7	r, x, z	a, b, c	
5,7,1	q, r, x, z	a, b, c	
5,7,1,6	d, q, r, x, z	a, b, c	er is geen regel die d waar kan maken
5,7,1	q, r, x, z	a, b, c	er is geen regel die q waar kan maken
5,7	r, x, z	a, b, c	er is geen regel die r waar kan maken
5	x, z	a, b, c	er is geen regel die x waar kan maken
5	y, z	a, b, c	
5,2	t, p, y, z	a, b, c	regel 8 vuurt, voegt t toe aan database
5, 2	p, y, z	a, b, c, t	
5, 2, 9	s, p, y, z	a, b, c, t	regel 3 vuurt, voegt s toe aan database
5, 2, 9	p, y, z	a, b, c, t, s	regel 9 vuurt, voegt p toe aan database
5, 2	y, z	a, b, c, t, s, p	regel 2 vuurt, voegt y toe aan database
5	z	a, b, c, t, s, p, y	regel 5 vuurt, voegt z toe aan database
		a, b, c, t, s, p, y, z	doel bewezen.

8. (a) (4 punten) Watson is een programma aan het debuggen. Zijn compiler is nu al een kwartier bezig met het compileren van het programma. Dat is lang voor een programma van deze omvang. Het valt Watson ook op dat de cursor in het outputscherf van de compiler is verdwenen en hij vraagt zich af of zijn compiler soms is blijven 'hangen'. Uit ervaring weet Watson dat zijn, niet al te betrouwbare, compiler eens in de 100 runs blijft hangen. Als dit het geval is dan verdwijnt de cursor in 60% van de gevallen. Als de compiler nog werkt, verdwijnt de cursor in 5% van de gevallen. Als de compiler inderdaad hangt dan zal deze na een kwartier of langer inderdaad geen reactie vertonen. Als de compiler nog wel werkt is er een kans van ongeveer 10% dat het compileren van het programma zeker een kwartier duurt. Bereken aan hand van de bovenstaande gegevens de kans dat de compiler van Watson toch nog werkt.

Antwoord:

W : compiler werkt nog.

C : cursor verdwenen.

R geen reactie na een kwartier.

Gegeven:

$$P(W) = \begin{array}{|c|c|} \hline false & 0.01 \\ \hline true & 0.99 \\ \hline \end{array} \quad (2)$$

$$P(C|W) = \begin{array}{|c|c|c|} \hline false & false & 0.40 \\ true & false & 0.60 \\ \hline false & true & 0.95 \\ true & true & 0.05 \\ \hline \end{array} \quad (3)$$

$$P(R|W) = \begin{array}{|c|c|c|} \hline false & false & 0.0 \\ true & false & 1.0 \\ \hline false & true & 0.90 \\ true & true & 0.10 \\ \hline \end{array} \quad (4)$$

Gevraagd: $P(W|C, R)$

Regel van Bayes:

$$P(W|C, R) = \frac{P(C, R|W)P(W)}{\sum_W P(C, R|W)P(W)} \quad (5)$$

We veronderstellen dat C en R conditioneel onafhankelijk zijn (gegeven W):

$$P(W|C, R) = \frac{P(C|W)P(R|W)P(W)}{\sum_W P(C|W)P(R|W)P(W)} \quad (6)$$

Invullen geeft:

$$P(W|C, R) = \frac{0.05 \times 0.10 \times 0.99}{0.05 \times 0.10 \times 0.99 + 0.60 \times 1.0 \times 0.01} \quad (7)$$

$$P(W|C, R) = \frac{0.00495}{0.00495 + 0.006} = 0.45 \quad (8)$$

- (b) (3 punten) Het al dan niet werken van de compiler kan, afgezien van de betrouwbaarheid van de compiler zelf, volgens Watson met twee dingen te maken hebben: er kan een bug in zijn programma zitten en het besturingssysteem kan hangen. Als dat laatste het geval is zou ook de webbrowser die Watson open heeft staan niet meer moeten reageren. Het niet of traag reageren van de browser kan natuurlijk ook te maken hebben met een trage internetverbinding.

Teken de structuur van een Bayesiaans netwerk waarmee Watson - als hij ook alle benodigde kansen zou weten - kan bepalen of zijn compiler hangt, neem hierin ook de informatie uit vraag (a) mee.

9. (2 punten) Stel we hebben de fuzzy sets A , B en C met membershipfuncties:

$$\mu_A(x) = (0.2/1, 0.3/2, 0.5/3, 0.7/4, 0.9/5) \quad (9)$$

$$\mu_B(x) = (0.9/1, 0.7/2, 0.4/3, 0.6/4, 0.8/5) \quad (10)$$

$$\mu_C(x) = (0.3/1, 0.6/2, 0.8/3, 0.6/4, 0.4/5) \quad (11)$$

Bereken de membershipfunctie van de fuzzy set $(\neg A \cup B) \cap C$ met behulp van de standaardoperatoren.

Antwoord:

$$\mu_{\neg A} = (0.8/1, 0.7/2, 0.5/3, 0.3/4, 0.1/5) \quad (12)$$

$$\mu_{\neg A \cup B} = (0.9/1, 0.7/2, 0.5/3, 0.6/4, 0.8/5) \quad (13)$$

$$\mu_{(\neg A \cup B) \cap C} = (0.3/1, 0.6/2, 0.5/3, 0.6/4, 0.4/5) \quad (14)$$

10. (a) (4 punten) Het perceptron is een van de meest eenvoudige neurale netwerken. Een perceptron leert met behulp van de update regel:

$$w_i(p+1) = w_i(p) + \alpha \times x_i(p) \times e(p). \quad (15)$$

Gegeven is een data set voor een netwerk met twee inputs en een output y :

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

en een netwerk met gewichten $w_1 = 0.2$, $w_2 = 0.1$ en drempelwaarde $\theta = 0.3$. Bereken de aanpassingen van de gewichten als elk voorbeeld 1 keer wordt aangeboden. Ga uit van een learning rate van 0.1.

Antwoord:

x_1	x_2	y_d	w_1	w_2	y	e
0	0	1	0.2	0.1	0	1
0	1	1	0.2	0.1	0	1
1	0	1	0.2	0.2	0	1
1	1	0	0.3	0.2	1	-1
			0.2	0.1		

- (b) (2 punten) Hoe werkt Hebb learning voor neurale netwerken?

Antwoord: Versterken van verbindingen tussen 2 neuronen als beide tegelijk hoge activatie. Een gebruikelijke methode is het vermenigvuldigen van de outputs van de neuronen $\Delta w_{ij} = \alpha y_j x_i$. Op deze manier kunnen gewichten alleen maar groter worden, om dit te voorkomen wordt soms een forgetting factor gebruikt.

- (c) (2 punten) Hoe werkt leren in een Kohonen neuraal netwerk (self-organising feature map)?

Antwoord: Leren in een Kohonennetwerk is unsupervised. De input wordt gezien als een vector $[x_1, x_2, \dots, x_n]$. Ieder neuron in output laag verbonden met ieder neuron in input laag. Deze verbindingen hebben een gewicht w_j , zodat ieder output neuron i een gewichtsvector $[w_{i1}, w_{i2}, \dots, w_{in}]$.

Deze vectoren worden vergeleken met de inputvector met behulp van een afstandsmaat (bijvoorbeeld Euclidische afstand). Het output neuron waarvan de gewichtsvector het dichtst bij de input is mag als enige zijn gewichten aanpassen. (In sommige varianten mogen ook een aantal neuronen in de omgeving van dit neuron hun gewichten aanpassen). Dit gebeurt meestal door een gewogen gemiddelde van de input en de oude gewichten te nemen $\Delta w_{ij} = \alpha[x_j - w_{ij}]$.

(d) (2 punten) Voor wat voor toepassingen zou je een Kohonen netwerk kunnen gebruiken?

Antwoord: Clusteren van data, patronen. (maken van een n -dimensionale afbeelding van een complexe ruimte).

11. (8 punten) We willen met behulp van genetic programming een formule bepalen die de nulpunten van kwadratische functies kan bepalen (ook wel bekend als de *abc*-formule). Leg uit hoe we dit kunnen doen. Behandel in je antwoord:

- terminals en primitieve functies
- de populatie
- de fitnessfunctie
- selectie
- cross-over
- mutatie
- de nieuwe populatie
- terminatie

Antwoord: De primitieve functies zijn wiskundige operatoren zoals, +, −, *, √, de terminals moeten in ieder geval de coëfficiënten a , b en c bevatten en constanten (1, 2, ...). De populatie bestaat uit N random gegenereerde formules opgebouwd uit de operatoren en de terminals (kunnen worden weergegeven als boomstructuren). De fitnessfunctie bestaat in dit geval uit een aantal voorbeelden waarvan we de nulpunten weten. Voor ieder chromosoom wordt de formule die het representeert toegepast op alle voorbeelden en de uitkomsten worden vergeleken met de correcte uitkomsten, bijvoorbeeld met behulp van de mean-squared error. Selectie vindt plaats met behulp van een roulettewiel. Dat wil zeggen dat de kans dat een chromosoom geselecteerd wordt evenredig is met de relatieve fitness. Bij cross-over worden delen van twee chromosomen uitgewisseld door in beide bomen een knoop te selecteren en deze inclusief de subbomen waar zij ouder van zijn om te wisselen. Mutatie komt neer op het vervangen van een functie door een willekeurige andere functie (met evenveel argumenten) of het vervangen van een willekeurig gekozen terminal door een andere terminal. Select, cross-over en mutatie worden herhaald totdat de nieuwe populatie even groot is als de oude. Dit wordt de nieuwe populatie in de volgende iteratie van het algoritme. Het algoritme stopt als een vooraf bepaald aantal iteraties is doorlopen of als de fitness voldoende hoog is (in dit geval de fout voldoende klein).

12. (3 punten) Leg uit hoe we gegevens kunnen clusteren met behulp van kunstmatige 'mieren' (ant-based clustering).

Antwoord: De gegevens worden verspreid over een 2 dimensionale matrix. De kunstmatige mieren bewegen zich door deze matrix. Als zij een datavector zien liggen is er een kans dat ze deze oppakken. Deze kans is groter als het aantal soortgelijke vectoren (bepaald door een afstandsmaat) in omliggende cellen klein is. Een mier die een datavector bij zich draagt legt deze neer met een kans die groter wordt naarmate het aantal soortgelijke elementen in omliggende cellen grote wordt.

Einde tentamenopgaven.

Controleer voor de zekerheid of je alle vragen hebt beantwoord. Het zouden er 12 moeten zijn.