

ARVOSTELUPERUSTEET

Kokeessa piti vastata viiteen (5) tehtävään kuudesta (6). Jokaisen tehtävän maksimipistemäärä on 8.

1. Tekoölyn filosofiaa yms.

a. (2 p) Selitä Turingin koe. (Huom. ei Turingin kone.) Minkälainen tekoöly on saavutettu, kun Turingin koe ratkaistaan?

– **koeasetelma (1p)**

– **saavutettu "ihmismäisesti toimiva" tekoöly (1p)**

b. (2 p) Elokuissa tekoöly usein saavuttaa tietoisuuden ja hyökkää ihmisten kimppuun. Herra Idionarap on huolissaan tästä riskistä ja ehdottaa lakialoitetta, jolla kaikki tekoölyalgoritmit kielletään. Millaisia käytännön seurauksia tällaisella lailla olisi? Mainitse muutama esimerkki. Mitä ongelmia lain valvontaan liittyisi?

– **Googlen, reittioppaan ja kameran kasvojentunnistuksen kaltaiset käytännön sovellukset eivät olisi sallittuja (1p)**

– **valvonta vaikeaa, koska "tekoölyalgoritmia" ei osata määritellä**

c. (2 p) Rouva Inoruen väittää, että tekoölymenetelmiä ei voi toteuttaa tavanomaisella tietokonelaitteistolla. Hänen mielestään siihen tarvitaan erityisiä neuroverkkoihin perustuvia laskennan malleja, joissa laskenta on rinnakkaista ja epäsynkronista. Onko rouva Inoruen oikeassa? Perustele.

– **ei ole oikeassa (1p)**

– **koska neurolaskentaa voidaan simuloida tavallisella tietokoneella (1p)**

d. (2 p) Minkälaisiin ongelmiin törmättiin 80-luvulla, kun tekoölyongelmia yritettiin ratkaista logiikan avulla? Miten näihin ongelmiin reagoitiin?

– **skaala, relevantin tiedon määrittely, epävarmuus ja epätarkkuus, ratkeamattomuus (1p)**

– **reagoitiin suuntaamalla rahoitusta neuroverkkojen ja todennäköisyyteen perustuvan tekoölyn tutkimukseen**

2. Etsintä ongelmanratkaisuna

Tehtävänä on ratkaista pulmatehtävä, jossa alkutila A on alla vasemmalla ja lopputila (maali) M on alla oikealla. Sallitut siirtymät tilojen välillä ovat sellaisia, että tyhjän ruudun vieressä (oikealla, vasemmalla, ylä- tai alapuolella) oleva numero siirtyy tyhjään ruutuun. Esimerkiksi alkutilasta voidaan siis siirtyä kumpaan tahansa tiloista 1 ja 2:

	+--++		+--++		+--++		+--++
	2 3		1 2		2 3		3
A:	+--++	M:	+--++	1:	+--++	2:	+--++
	1		3		1		2 1
	+--++		+--++		+--++		+--++

- a. (2 p) Esitä kaikki alkutilasta saavutettavat tilat tilakaaviona. Merkitse kaavioon myös tilojen väliset siirtymät.

– vastaukseksi hyväksyttiin tilakaavion (jossa jokainen tila esiintyy tasan kerran) lisäksi etsintäpuu

- b. (3 p) Simuloi sekä syvyysuuntaista että leveysuuntaista hakua a-kohdan tilasiirtymäkaaviossa. (Riittää luetella tilat siinä järjestyksessä, kun ne käydään läpi.) Kuinka monta tilaa kumpikin hakumenetelmä käy läpi ennen kuin maalitila löytyy? Kumpi löytää paremman ratkaisun (vähemmän siirtoja)? Onko tulos aina sama riippumatta siitä, missä järjestyksessä tiloja tarkistetaan?

syvyysuuntainen:

$A=[23_1]-[231_]-[2_13]-[_213]-[12_3]-[123_]=M$, joka on myös syvyysuuntaisen palauttama reitti (5 siirtymää), jos haku lähtee tähän suuntaan; toiseen suuntaan lähteminen tuottaa 7 siirtymää pitkän reitin

leveysuuntainen:

$A=[23_1]-[_321]-[2_13]-[3_21]-[_213]-[312_]-[12_3]-[31_2]-[123_]=M$. Etsintä tuottaa saman reitin kuin syvyysuuntainen parhaimmillaan (5 siirtymää).

pisteitä sai siitä, että oli osannut

– luetella tilat oikeassa järjestyksessä (1p) ja

– antaa kummankin etsintäalgoritmin tuottaman reitin (1p)

– kertoa, että leveysuuntainen haku löytää aina reitin, jossa on minimimäärä siirtymiä (1p)

- c. (3 p) Kuvitellaan että tehtävänä olisi ohjelmoida Reittiopas, joka annettuna mitkä tahansa julkisen liikenteen pysäkit A ja B, viikonpäivä ja kellonaika, etsii nopeimman reitin A:sta ja B:hen käyttäen hyödyksi tietoa julkisen liikenteen reiteistä ja aikatauluista.

Miten lähtisit ratkaisemaan tehtävää käyttäen hyödyksi etsintäalgoritmeja. Mikä olisi etsinnän “tilan” esitys ja mitkä mahdolliset tilasiirtymät? Minkälainen hakualgoritmi sopisi tehtävään? Käyttäisitkö jonkinlaista heuristiikkaa, ja jos käyttäisit, mitä?

- **tila: [pysäkki, klo], siirtymät: reitit (1p)**
- **algoritmi A* (1p)**
- **heuristiikka: linnuntie-etäisyys jaettuna maksiminopeudella (1p)**

Myös joistain muista toimivan kuuloisista ratkaisuista saa pisteitä.

3. Kuvankäsittely

a. (3 p) Miksi digitaaliset signaalit kuten kuva ja ääni tuottavat ongelmia logiikkaan perustuville ja muille GOFAI (“Good Old Fashioned AI”) -menetelmille? Millaiset menetelmät soveltuvat digitaalisten signaalien käsittelyyn paremmin ja miksi?

- **ongelmia: koko, kohina, vaikea määritellä mitä tarkoitetaan tietyllä hahmolla, ... (1p)**
- **modernin AI:n menetelmät, kuten koneoppiminen, digitaalisen signaalinkäsittelyn menetelmät, todennäköisyysmallinnus, ... (1p)**

b. (3 p) Jos tehtävänä olisi toteuttaa kasvojentunnistusmenetelmä, joka tunnistaa kameran avulla kenestä henkilöstä on kyse, minkälaisen menetelmän valitsisit? Selitä lyhyesti, miten valitsemasi menetelmä toimii. Toimisiko menetelmä aina vai olisiko sillä jotain tiettyjä rajoitteita?

– **esim. invariantteihin piirteisiin perustuvat SIFT ja SURF:**

1. **avainpisteiden valinta tutkimalla pisteiden ympäristön pikseleiden intensiteetin vaihteluita; löytyy "blobeja"**

2. **avainpisteiden ympäristöä kuvaavan piirrevektorin muodostaminen**

3. **kahdesta eri kuvasta muodostettujen piirrevektorien vertailu**

(2p)

rajoitteet: ei toimi, jos olosuhteet muuttuvat liian rajusti, ei siedä kuvakulman muutoksia, jne (1p)

c. (2 p) Ohessa on kohinanpoistoalgoritmin pseudokoodi. Selitä lyhyesti miten algoritmi toimii ja mikä sen taustalla oleva idea on.

kohinanpoisto(x,t):

n ← length(x)

c ← dwt(x) // aallokemuunnos (discrete wavelet transform)

for i = 1,...,n:

if |c[i]| < t: c[i] = 0

```
end-for
x ← idwt(c)      // käänteismuunnos (inverse wavelet transform)
return x
```

– pseudokoodin selitys oikein (1p)

– **idea: varsinainen signaali koostuu harvoista isoista kertoimista ja isosta joukosta lähellä nollaa olevia kertoimia ja kohina jakautuu kaikille kertoimille tasaisesti => kun nollataan lähellä nollaa olevat, signaalista häviää pieni osa, mutta kohinasta iso osa (1p)**

4. Neuroverkot

a. (2 p) Oikein vai väärin?

i. Monikerrospereptronissa neuronit kilpailevat siitä, mikä niistä saa aktivoitua.

– **väärin: kilpailua tapahtuu Kohosen kartassa**

ii. Takaisinkytketyn neuroverkon avulla voidaan toteuttaa vikasietoinen muisti.

– **oikein**

iii. Bayes-verkko on esimerkki stokastisesta (satunnaisuuteen perustuvasta) neuroverkosta.

– **väärin: Bayes-verkko on todennäköisyysmalli, ei neuroverkko**

iv. Keinotekoisien neuroverkkojen neuroneihin liittyy painokertoimet, jotka yhdessä neuronin syötteiden kanssa määräävät aktivaation.

– **oikein**

pisteet:

0-2 oikein (keskimäär. tulos arvaamalla) = 0p, 3 oikein = 1p, 4 oikein = 2p

b. (2 p) Kuvaile valitsemasi neuroverkon (*) toimintaperiaate. Mitkä ovat verkon syötteet ja mitä tapahtuu, kun syöte esitetään verkolle?

– **arvostelu tapauskohtaisesti, esim. tyyliin SOMissa neuronit kilpailevat siitä, mitä niistä vastaava painovektori on lähinnä syötettä. Tämä neuronit aktivoituu syötteestä.**

c. (2 p) Selitä kyseisen verkon oppimismenetelmä. Riittää selittää toimintaperiaate, ei tarvitse esittää yksityiskohtia. (Ei pseudokoodia, vaan on selitettävä sanallisesti.) Minkä muotoista opetusaineistoa verkon opettamiseen käytetään?

– **arvostelu tapauskohtaisesti, esim. tyyliin Aktivoituneen SOM-neuronin painovektoria muokataan siten, että se on entistä lähempänä syötevektoria; myös naapureiden vektoreita muokataan samaan suuntaan. Syötteenä on vektoreita.**

d. (2 p) Mikä on tyypillinen kyseisen neuroverkon sovellus?

– arvostelu tapauskohtaisesti, esim. tyyliin SOMin tyypillinen sovellus on datan visualisointi

*) Huom: Kuvailtavassa verkossa on oltava useampi kuin yksi yhteenliitetty neuroni.

5. Todennäköisyysmallinnus

Ohessa muutamia todennäköisyyslaskennan kaavoja muistin virkistykseksi:

- (i) $P(\neg A) = 1 - P(A)$ "negaatio"
- (ii) $P(A \vee B) = P(A) + P(B) - P(A, B)$ "disjunktio"
- (iii) $P(A) = P(A, B) + P(A, \neg B)$ "marginalisointi"
- (iv) $P(A_1, \dots, A_k) = P(A_1) P(A_2 | A_1) P(A_3 | A_1, A_2) \dots P(A_k | A_1, \dots, A_{k-1})$ "ketjusääntö"
- (v) $P(A | B) = P(A) P(B | A) / P(B)$ "Bayesin kaava"
- (vi) $A \perp B | C \Rightarrow P(A, B | C) = P(A | C) P(B | C)$ "ehdollinen riippumattomuus"

a. (2 p) Olkoon $P(V | S) = 0.25$ ja $P(V | \neg S) = 0.0005$, missä S tarkoittaa, että viesti on roskapostia (spam) ja V tarkoittaa, että viestissä esiintyy sana "Viagra". Laske todennäköisyys $P(S | V)$, kun $P(S) = 0.2$. Voit antaa lopputuloksen rationaalilukuna eli muodossa r/s, missä r ja s ovat molemmat kokonaislukuja.

– Bayesin kaava.

$$0.2 \times 0.25 / (0.2 \times 0.25 + 0.8 \times 0.0005) = 125/126 \sim 99.2 \%$$

(2p) lasku- tai muista huolimattomuusvirheistä ei sakota

b. (2 p) Seuraavassa taulukossa on laskettuna joidenkin sanojen esiintymistiheys sekä roskapostissa että asiallisissa viesteissä (ham).

sana	spam	ham
<i>money</i>	24	6
<i>lottery</i>	16	4
<i>Bayes</i>	1	10
yhteensä	2000	10000

Arvioi todennäköisyydet $P(\text{Sana}_i = \textit{money} | \text{spam})$, $P(\text{Sana}_i = \textit{lottery} | \text{spam})$ ja $P(\text{Sana}_i = \textit{Bayes} | \text{spam})$ sekä vastaavat todennäköisyydet hammille.

– f/n , missä f esiintymät ja n ko. luokan sanojen määrä yhteensä

24/2000, 16/2000, 1/2000, 6/10000, 4/10000, 10/10000

(2p) huom: ei tarvittu Bayesin kaavaa tai muutakaan laskusääntöä f/n -kaavan lisäksi.

- c. (4 p) Laske todennäköisyys, että viesti on roskapostia, kun sen sisältö on "Bayes lottery money". Käytä naivi Bayes -mallia, jossa viestin sanat ovat riippumattomia toisistaan, kun viestin tyyppi (spam/ham) on annettu. Käytä tässäkin prioria $P(\text{spam}) = 0.2$. Tuloksen voi taas antaa rationaalilukuna.

– kannatti (muttei ollut pakko) laskea ensin

$$\text{odds} = \frac{P(\text{spam}) P(\text{Bayes}|\text{spam}) P(\text{lottery}|\text{spam}) P(\text{money}|\text{spam})}{(P(\text{ham}) P(\text{Bayes}|\text{ham}) P(\text{lottery}|\text{ham}) P(\text{money}|\text{ham}))}$$

$$= 0.2/0.8 \times 1/2000 / (10/10000) \times 16/2000 / (4/10000) \times 24/2000 / (6/10000) = 50$$

(3p)

ja muuntaa sitten

$$P(\text{spam}|\text{viesti}) = \text{odds}/(1+\text{odds}) = 50/51 \sim 98.0 \% \text{ (1p)}$$

lasku- tai muista huolimattomuusvirheistä ei sakota

6. Robotiikka

Koska Legojen rakennus on hauskaa, tässä tehtävässä pääset suunnittelemaan Lego-robotin ja sen ohjelman!

Alla on kuvia robotin osista:

- ▶ keskusyksikkö, jossa on portit kahdelle moottorille (M1,M2) ja kolmelle sensorille (S1,S2,S3),
- ▶ kolme pyörää,
- ▶ kaksi moottoria (joita voi ohjata erikseen pyynnöillä *forward()*, *backward()*, *stop()*),
- ▶ ultraäänisensori (joka palauttaa pyynnöllä *getDistance()* etäisyyden lähimpään esteeseen (cm)),
- ▶ valosensori (joka palauttaa pyynnöllä *getLightValue()* sensorin alla olevan lattian vaaleutta kuvaavan arvon)



Lisäksi saat tietysti käyttää normaaleja Lego-palikoita ja akseleita, kaapeleita ja muita tarvittavia osia.

- a. (2 p) Piirrä yksinkertainen robotti, jolla voit ratkaista seuraavan kohdan tehtävän. Piirrä kuvaan myös kaapelit, joilla yhdistät oikeat komponentit keskusyksikön portteihin.

– **tapauskohteisesti (2p)**

- b. (4 p) Esitä pseudokoodina ohjelma, joka seuraa lattiaan piirrettyä mustaa viivaa niin kauan, että tulee ultraäänisensorilla havaittavan esteen luokse ja pysähtyy sen jälkeen. Voit olettaa että robotti asetetaan aluksi mustan viivan päälle.

Ohjeita: Komento *M1.forward()* saa porttiin M1 kytketyn moottorin liikkeelle (eteenpäin). Sensoreilta voit kysyä arvoja komennolla *S1.getDistance()* jne. Moottorien ja sensorien konstruktoreja tai muita alustuksia ei tarvitse esittää.

– **tapauskohteisesti:**

perusidea järkevä (3p)

pääosin toimiva toteutus; pari pikkuvirhettä sallitaan (1p)

- c. (2 p) Miksi paperilla suunniteltava robotin ohjelma ei yleensä toimi ensi yrityksellä? Mainitse joitakin oikeaan fyysiseen ympäristöön liittyviä asioita, jotka aiheuttavat ylimääräisiä pulmia verrattuna “tavalliseen” (ei-robotti-) ohjelmointiin ja kuvaile, miten kyseiset pulmat voitaisiin ratkaista.

– **ei toimi johtuen ympäristön odottamattomista ominaisuuksista, sensorihavaintojen häiriöistä ja epätarkkuudesta sekä liikkumisen epätarkkuuksista ja siitä, että on "vaikea ajatella kuten robotti" (lainatakseni yhtä vastausta). (1p)**

– **voidaan ratkaista testaamalla ja korjaamalla: sensoreille raja-arvot tämän perusteella, jne. (1p)**