

Institutt for Datateknikk og Informasjonsvitenskap

Eksamensoppgave i TDT4137 Kognitive Arkitekturer

Faglig kontakt under eksamen: Asbjørn Thomassen

Tlf.: 73591839

Eksamensdato: 10 desember 2015

Eksamenstid (fra-til): 0900-1300

Hjelpemiddelkode/Tillatte hjelpemidler: D/ Kalkulator

Annen informasjon:

Målform/språk: bokmål

Antall sider (uten forside): 4

Antall sider vedlegg: 1

Kontrollert av:

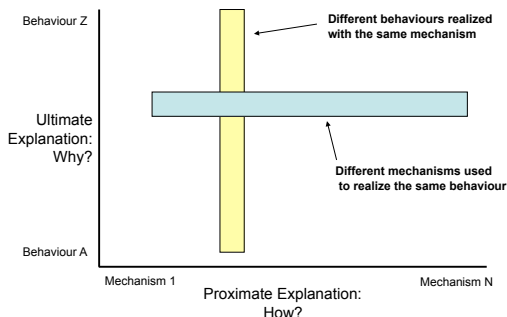
Dato

Sign

Oppgave 1 (20%)

a) Hva er ultimate-proximate-distinksjonen relatert til kognitive systemer?

Svar:



(Er et av de fire aspekt for modellering av kognitive systemer, se Vernon intro)

b) I modellering av kognitive systemer har Marr og Kelso to forskjellige hierarkiske syn. Redegjør kort for forskjellene. Marr: top level viktigst, Kelso: alle nivåer like viktig (embodied)

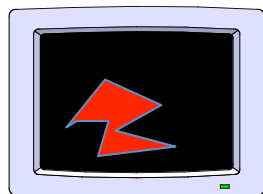
Svar: Marr's hierarchy:

- The problem can, and should, be modelled first at the level of the computational theory without reference to the lower levels less abstract levels
- Then consider the design options (algorithm and representation)
- Then consider the implementation options

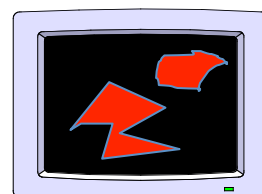
Kelso's hierarchy:

- Take away the context, and you take away the basis for the model
- Instantiation of the system has a direct role to play in the model itself
- Embodiment is integral to the model

c) Figuren nedenfor til venstre viser en skjerm med et ukjent grafisk bilde. Anta at dette allerede er representert i WM. Enda et nytt ukjent grafisk symbol dukker nå opp på skjermen, se bilde til høyre. Hvor lang tid tar det i MHP (Model Human Processor) å avgjøre om symbolene er like? Begrunn svaret.



Første symbol på skjermen



Begge symboler på skjermen

Svar: $T_p + T_c$ (andre symbol tas inn i WM + match av visuell representasjon av symbolene)

d) Fitts lov kan betraktes som utførelsen av n kontrollerte mikro-bevegelser. Hva skjer i hver mikrobevegelse i lys av MHP? **Svar:** En ser(P), gjør korreksjon(P), genererer ny mikrobevegelse (kontrollert feedback loop).

Hva blir tiden for å utføre n mikrobevegelser (tiden som funksjon av n)?

Svar: $T=n(T_p+T_c+T_m)$

e) Hva sier Rasjonalitetsprinsippet (i MHP)?

Svar:

"A person acts so as to attain his goals through rational action, given the structure of the task and his inputs of information and bounded by limitations of his knowledge and processing ability:

Goals + Task + Operators + Inputs+ Knowledge + Processing Limits ->Behavior"

En rasjonal aksjon sees på som en bevegelse i et tilstandsrom (problem space):

"Hvis en agent har kunnskap om at anvendelsen av en operator vil lede til et av dets mål, så vil agenten velge denne operatoren" (state - operators-constraints, and control knowledge).

Nevn en annen arkitektur som også er basert på dette prinsippet.

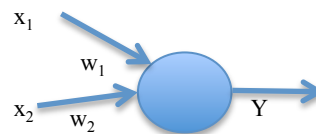
Svar: Soar og Icarus er basert på denne

Oppgave 2 (20%)

a) Hvordan modelleres soma, synapse, axon og dendritt som et kunstig neuron (node) i et nevralt nett? **Svar:** soma: hele neuronet, axon: output, dendrite: input, synapse: vekt

b) Vi skal lære opp et perseptron til å representere den logiske funksjonen AND. Perseptronet har to input variable x_1 og x_2 med tilhørende vekter w_1 og w_2 . Y_d , desired "output" for AND, er gitt ved:

X_1	X_2	Y_d
0	0	0
0	1	0
1	0	0
1	1	1



Output beregnes med en step-aktiveringsfunksjon:

$$Y(p) = \text{step} \left[\sum_i x_i(p) w_i(p) - \theta \right]$$

Funksjonen $\text{step}(X)$ er 0 hvis $X < 0$ og 1 ellers. Læring gjøres med deltaregelen:

$$\Delta w_i(p) = \alpha x_i(p) e_i(p)$$

Hva er θ , $e_i(p)$ og α ? Hva er ideen bak deltaregelen?

Svar: θ er threshold, $e_i(p)$ er feilen i iterasjon (runde) p er definert ved: $e_i(p) = Y_d - Y$ og α er læringshastigheten. Deltaregelen uttrykker en gradient nedstigning langs feil-"flaten" $E = \frac{1}{2}(Y_d - Y)^2$ for å finne vektene som minimalisere feil (helst globalt minimum, men kan ende i et lokalt).

- c) Anta at vi har startet AND-treningen av perseptronet med $\theta = 0.2$ og $\alpha = 0.1$ og har fått følgende verdier på vektene $w_1 = 0.2$ $w_2 = 0.1$.

Vis hvordan vektene oppdateres med i ny runde med treningsdata.

Svar: For hver input: beregn Y som gitt ovenfor. Hvis feil output: endre vektene med deltaregelen (p er iterasjon nr p):

$$w(p+1) = w(p) + \Delta w(p)$$

Når det ikke lenger er "feil": er du ferdig.

Dette gir følgende resultat med $\theta = 0.2$ og $\alpha = 0.1$ (må bruke formelen for hver input, noe som gir fire utregninger):

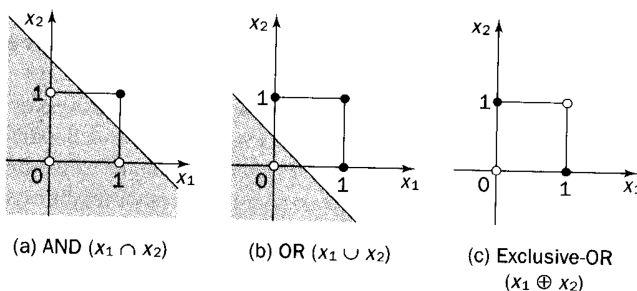
x_1	x_2	Y_d	w_1	w_2	Y	e	nye vekter: w_1	w_2
0	0	0	0.2	0.1	0	0	0.2	0.1
0	1	0	0.2	0.1	0	0	0.2	0.1
1	0	0	0.2	0.1	1	-1	(må endre w her) 0.1	0.1
1	1	1	0.1	0.1	1	0	0.1	0.1

Er perseptronet nå ferdigtrønt?

Svar: Vi har fremdeles en feil $e = -1$ i epochet (datasettet), dvs **ikke** ferdigtrønt. Må kjøre en eller flere "runder" til.

- d) Hva er lineær separabilitet, og hva betyr det for perseptronet? Gi et eksempel.

Svar: Datasettet kan deles i to klasser vha en rett linje (i 2D), dvs lineær funksjon. Ser vi på ligningen for perceptronet, er den basert på en lineær kobinasjon av input og vekter. Dette betyr at perceptronet bare kan lære lineære funksjoner (behøver derfor multilayer networks for å takle ikke-lineæraritet). Eksempel f.eks:



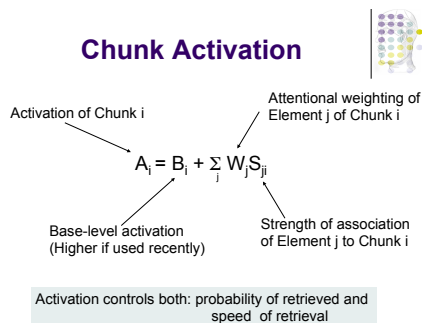
Ser her at både AND og OR er lineær separable, mens XOR ikke lar seg skille med en rett linje (0ere på den ene siden, og 1ere på den andre siden av linja)

- e) ACT-R har et subsymbolsk nivå for bla. aktivering av chunker. For en chunk i , kan dette uttrykkes ved:

$$A_i = B_i + \sum_j W_j S_{ji}$$

Forklar kort hva de ulike leddene beskriver.

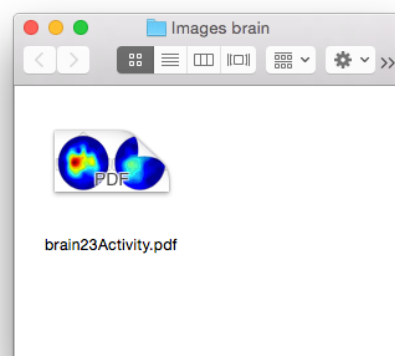
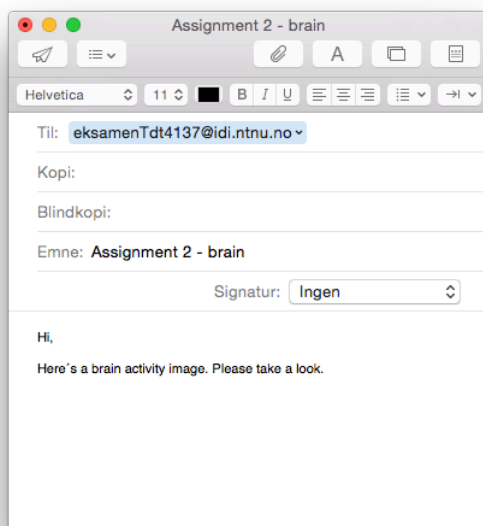
Svar: B_i er base-level (egenaktivering) og summen gir kontekst-aktivering (av chunker relatert til i)



Oppgave 3 (20%)

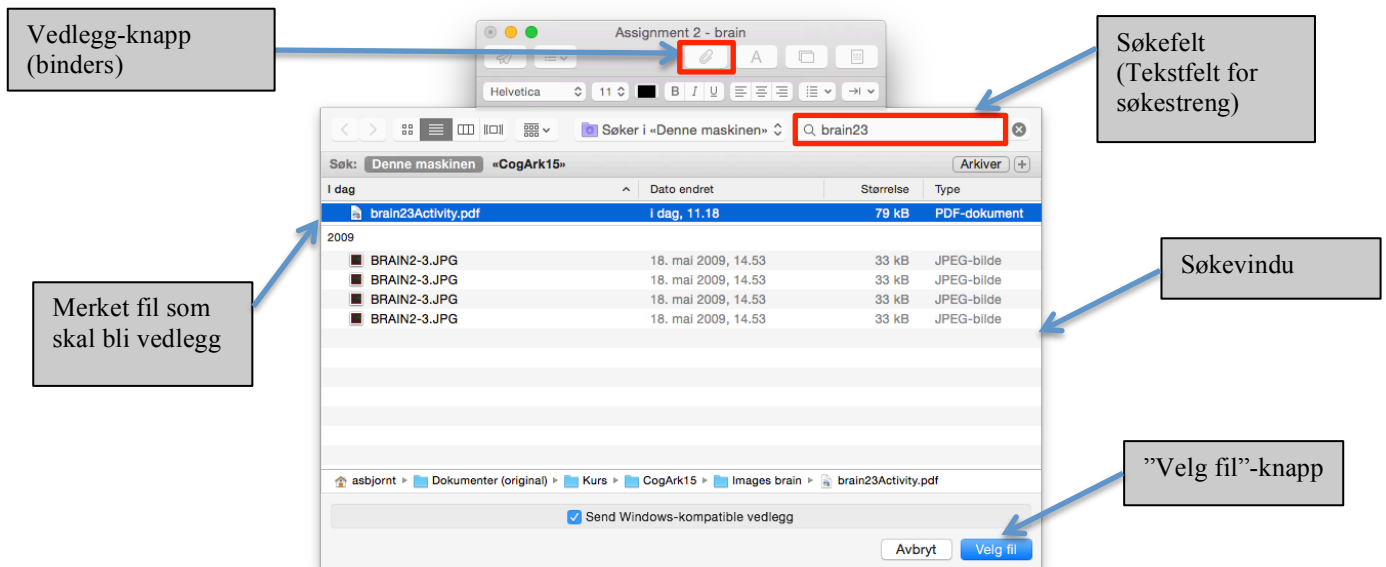
I denne oppgaven skal vi analysere en del av en mail-applikasjon med GOMS. Anta at du skal sende et aktivitetsbilde av hjernen på mail til en eller annen person (hvem er uvesentlig her ☺). Du har åpnet en ny mail og skrevet en liten følgetekst i tekstfeltet, som vist i figuren nedenfor. Du skal nå **legge til bildet som vedlegg** og er her begrenset til å gjøre dette på to alternative måter:

Alternativ 1: Bildefilen du skal legge ved er **synlig** i et eget mappe-vindu. Figuren under viser skjermen med to synlige vinduer: mailvinduet til venstre og mappen "Images brain" (fil-vinduet) som inneholder bildefilen "brain23Activity" som skal sendes med mailen. Siden bildefilen er synlig, kan den her bare dras over til mailvinduet for å bli vedlegg.



Du må da først klikke i fil-vinduet for å gjøre det aktivt, og deretter gjøre drag-and-drop til mail-vinduet. Etterpå må du igjen klikke i mail-vinduet for å komme tilbake til mailen igjen.

Alternativ 2: Dersom bildefilen **ikke er synlig** kan vedleggs-knappen, dvs "bindersen" rett under vindusnavnet "Assignment 2 – brain", trykkes på for å få opp et søkevindu på skjermen:



Anta at du bruker søkefeltet i søkevinduet for å finne bildet, se figuren ovenfor. Dette gjør du ved at du først klikker i feltet og så skriver inn (deler av) filnavnet. Anta at du sjekker for hver bokstav om filen dukker opp i søkevinduet. Når ønsket fil er funnet kan denne merkes ved å klikke på linjen. Linjen med filen blir da "uthevet". Ved deretter å trykke på knappen "Velg fil", blir bildet vedlegg til mailen.

- Hva er en unit-task? Gi eksempler fra mail-applikasjonen. Skriv unit-task nivået for mail-applikasjonen i NGOMSL.
- Betrakt oppgaven "legg til vedlegg" beskrevet tidligere i oppgaven. Skriv NGOMSL-modellen for de to alternativene.
- Beregn utførelstid for Alternativ 1.
(CP = 1.2, B=0.1, H=0.4, K=0.2, M=1.2, P = 1.1 - alt i sekunder)
- Hva er hovedskillet mellom KLM (Keystroke Level Model) og NGOMSL-modellering? Gi et par eksempler på hvordan den underliggende arkitekturen (MHP) kommer til syne i NGOMSL.

Svar a) og b):

a) Top-level nivå med unit-task håndtering (hva du kan gjøre i mail-appen): hent mail, lag ny mail, lag ny postboks, forward mail, legg til vedlegg,...slett mail, etc.

Method to accomplish goal: *handle mail*

Step 1. Get next unit task

Step 2. Decide: If no more unit tasks, then return with goal accomplished

Step 3. Accomplish the goal: *performing the unit task*

Step 4. Goto 1

Selection rule set for the goal: *performing the unit task*

If task is add enclosure then accomplish: *addEnclosure*

If task is to send mail accomplish: *sendMail*

If task is make new mail then accomplish: *makeNewMail*

If task is open mail then accomplish: *openMail*

If task is create new mail box then accomplish: *createMailBox*

...

Return with goal accomplished

b)

Videre modelleres bare målet *addEnclosure*:

Siden det er to måter å legge til vedlegg, får vi en seleksjonregel (Selection rule set). Viktig.

Selection rule set for the goal: *addEnclosure*

If task is add enclosure by drag and drop then accomplish: *dragandDropFilefromWindow*

If task is add enclosure by search: *addBySearchWindow*

Return with goal accomplished

Method for goal: *dragandDropFilefromWindow*

Step 1. Locate file in window

Step 2. Move cursor to file in window

Step 3. Click to activate window

Step 4. Drag file to mail-window // uses generic method "drag object to destination"

Step 5. Return with goal accomplished

Method for goal: *drag object to destination*

Step 1. Press mouse button on object

Step 2. Verify object inverse

Step 3. Locate destination

Step 4. Move object to destination

Step 5. Release mouse

Step 6. Return with goal accomplished

Method for goal: *addBySearchWindow*

Step 1. Accomplish goal: *push "paper clip" button* //use generic button method

Step 2. Accomplish goal: *searchAndAddFile*

Step 3. Return with goal accomplished

Method for goal: *push button*

Step 1. Locate button

Step 2. Move cursor to button

Step 3. Press button

Step 4. Verify feedback

Step 5. Return with goal accomplished

Method for goal: *searchAndAddFile*

Step 1. Move cursor to search field

Step 2. Click

Step 3. Accomplish goal: *nameSearch*

Step 4. Move and Click to mark

Step 5. Accomplish goal: *push "Velg fil" button*

Step 6. Return with goal accomplished

Method for goal: nameSearch
Step 1. Recall file name
Step 2. Type next letter
Step 3 . Locate file
Step 4. If file not found go to step 2
Step 5. Return with goal accomplished

c) Totaltid er $T = \sum \text{KLM-tider} + \text{antall step} \cdot 0.1 \text{ sek}$ (bare se at de forstår dette, dvs teller opp antall step i programtracen og legger til KLM-operatortider. ikke så farlig med estimatet).

d) KLM kan sees på som et mentalprogram uten subrutiner og parametre (hovedprogram). Behandler ikke bruk av hukommelse (internt). Bare konkrete task-scenariorer. NGOMSL har subrutiner, generalisert bruk og modellerer bruk av hukommelse bla (modellerer intern kognitiv prosessering).

Arkitekturen kommer til syne gjennom operatører som: forget, retrieveLTM, retainWM og ellers også i oppbyggingen av seleksjonsreglene (parallell produksjonssystem-*if*).

Oppgave 4 (20%)

I et dataspill skal en agent (robot) vurdere en trussel fra mulige inntrengere i et landskap. Agenten bruker fuzzy-resonnering for å finne ut hvor mange forsvarere som er nødvendig å sette ut i landskapet. Antallet forsvarere bestemmes ut fra to lingvistiske variable *range* og *forceSize*. *range* er avstanden til inntrengerne og opererer på fuzzy-settene: close, medium, far. *forceSize* angir inntrengernes troppestørrelse og har settene: Tiny, Small, Moderate, Large. Agentens trusselvurdering beskrives ved output variabelen *threatLevel* som har settene: Low, High og Medium. *threatLevel* gir antall forsvarere som bør settes ut. Alle settene med regler er beskrevet i Vedlegg 1.

- a) Gitt reglene og innverdier for *range* på 25 og *forceSize* på 8.
Hvilken trusselvurdering gjør agenten?
Anta Mamdani-resonnering og vis trinnene i hvordan du kommer frem til svaret. Du kan skrive og tegne direkte på Vedlegg 1.

Svar:

1. Fuzzification (gir verdiene som vist i fuzzy settene i vedlegg 1)
2. Rule evaluation (gir verdier som vist over reglene i vedlegg 1)
3. Aggregation (se svart sett tegnet inn på action-settet i vedlegg 1)
4. Defuzzication. Mamdani bruker tilnærmer integrasjon med summering i step på 5

Se vedlegg 1 for utregning

- b) Hva skiller Mamadani fra Sugeno?

Svar: Mamdani beskriver mer human-like spesifikasjon av regler (fanger kunnskap lettere), men er tyngre å beregne enn Sugeno. Sugeno er lett å beregne pga singletons og egner seg derfor til spesielt til bl.a kontroll systemer (ikke-linære systemer)

Skriv om reglene i vedlegg 1 til Sugeno ved at du velger singletons i 10, 30 og 50 for *threatLevel*. Vis hvordan du nå beregner trusselvurderingen.

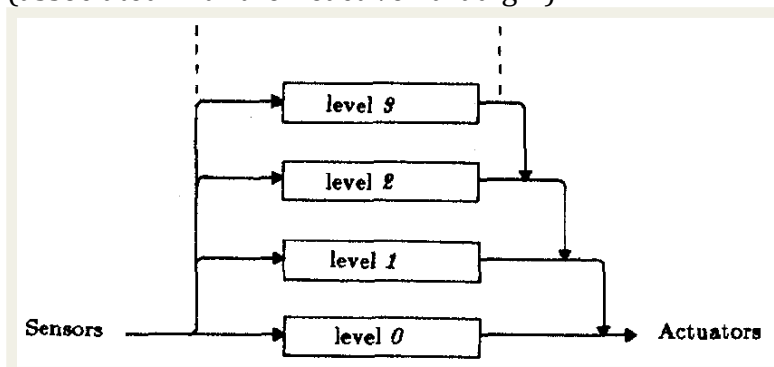
Svar: se vedlegg 1 for utregning (bare 3 og 4 i a) er annerledes ved at en bruker singeltons i stedet for sett. Defuzz. er vekta gjennomsnitt.

Oppgave 5 (20%)

- a) Hvordan representeres en tilstand i Icarus? **Svar:** Som et sett av beliefs.
Hva er en impasse? **Svar:** Icarus kan ikke finne et skill (feiler) for å løse et goal og må gå over til problemløsning. (Skill er en metode/prosedyre som kan være hierarkisk og som driver Icarus mot en måltilstand).

Forklar kort hvordan Icarus gjør problemløsning. **Svar:** når condition delen i et skill feiler, prøver en å løse dette ved å se etter **skills som matcher ikke-opnådde mål i condition**-delen. Dersom en ikke finner et skill, prøver en å finne et **konsept** som ikke er oppfylt og utføre andre skills for oppfylle konseptet for så å komme videre (beliefs).

- b) Beskriv kort Soars prosesseringssyklus (processing cycle) fra agenten får input til den evt. genererer output. **Svar:** Viser her til foiler og artikkel (input, elaborate, decision, application, output).
Hva er forskjellen på ($\langle s \rangle \hat{\text{operator}} \langle o \rangle$) og ($\langle s \rangle \hat{\text{operator}} \langle o \rangle +$)?
Svar: Den første er automatisk generert av Soar og viser valgt operator som skal utføres. Den andre er et forslag til operator (generert av Soar-programmet vårt) med "acceptable" preferanse. Soar velger den operatoren som har høyest preferanse blant alle forslagene.
- c) Hva er hoved-ideen bak "Brooks subsumption architecture"?
Svar: Vertical decomposition of tasks into a Sense-Action Organisation (levels) (associated with the Reactive Paradigm).



A behaviour is a network of sensing and acting modules which accomplish a task. The modules are Augmented Finite State Machines (AFSM), or finite state machines which have registers, timers & other enhancements to permit them to be interfaced with other

modules. Modules in a higher layer can **override or subsume** the output from behaviours in the next lower layers.

Outputs are simple functions of inputs and local variables

Hvilken rolle spiller *inhibit* og *suppress* i arkitekturen? **Svar:**

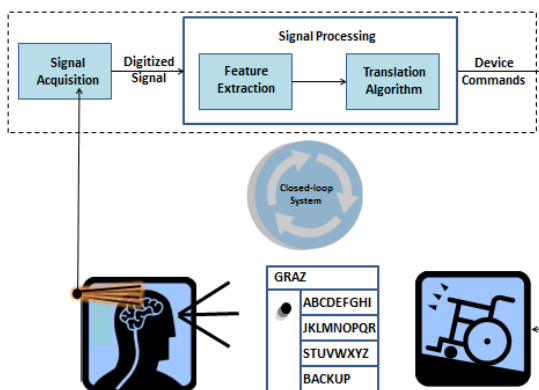
Inputs can be suppressed and outputs can be inhibited. Suppress: replace input.

Inhibit: all output lost for period of time.

This function allows higher levels to subsume the function of lower levels

Lower, therefore, still function as they would without the higher levels

d) Forklar kort de fire komponentene i BCI (Brain Computer Interaction). Tegn gjerne opp.



signal acquisition (get information from brain)

signal processing (translate information to messages or commands)

devices and applications (speller, robotic device, etc)

application interface (or **operating environment**) determines how components interact (with each other and the user)

Hvorfor gjøres windowing på EEG-signalet før Fast Fourier Transform (FFT)?

Svar: EEG-signaler er ikke-periodiske. Fourier kan bare gjøres på periodiske funksjoner. Får ellers problemer med frekvenslekkasje (leakage). Windowing gjør EEG-signalet periodisk.

Hva er mu-bølger og P300? Gi eksempler på hva de kan brukes til.

Svar (krever ikke så mye her): mu-bølger: The sensorimotor cortex has a salient 10 Hz rhythm ("mu") which disappears during voluntary movement. Kan brukes til styring av kunstige lemmer (disabled). . A person suppresses mu wave patterns when he or she performs a motor action or, with practice, when he or she visualizes performing a motor action.

P300 = positively deflected peak in the raw EEG signal (also called P3b). Reflects processes involved in stimulus evaluation or categorization. occurs approximately 300 milliseconds after the presentation of an unexpected target stimulus. (low-probability target items are mixed with high-probability non-target (or "standard") items. (Odd-ball paradigm, typically visual, auditory, or somatosensory). Metric for cognitive function in decision making processes (også løgndetektor).

- e) I et ASR-system (Automatic Speech Recognition) skal du velge setningen W som "matcher" observasjonsvektoren X best, dvs beregne $\operatorname{argmax} P(W|X)$. Anta at du har en språkmodell (Language model) som gir $P(W)$ og en akkustisk prosessering som gir $P(X|W)$.
Vis hvordan du beregner $P(W|X)$.

Svar:

$$P(W|X) = \frac{P(X|W) P(W)}{P(X)} \quad (\text{etter Bayes' Regel})$$

$$\operatorname{argmax} P(W|O) \approx \operatorname{argmax} P(X|W) P(W)$$

siden $P(X)$ er den samme for alle ordsekvenser

Typiske dekodere : Viterbi eller A^* (finner en vei (ordsekvens) med størst sannsynlighet)

Hva er tanken bak bruk av N -grams i språkmodellen?

Svar: (se foiler)

0.7
THEN threatLevel is Low

0.15 0.7 0.15
IF range is Close AND forceSize is Tiny THEN threatLevel is Medium

0.15 0.7 0.15
IF range is Close AND forceSize is Small THEN threatLevel is High

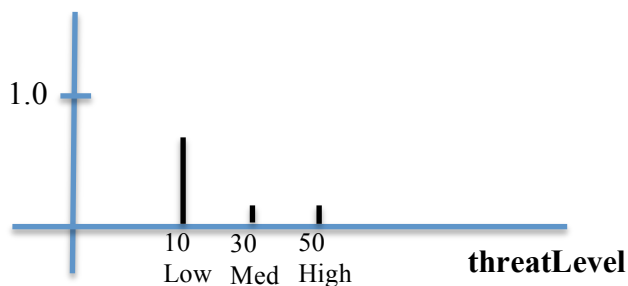
Regner ut COG (Mamdani)

$$COG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx}$$

– tilnærmer integrasjonen med step på 5 fra 5 til 60 langs x (finner areal under svart kurve)
= 0.7(5+10)+0.5*15+20*0.3+(25+30+35+40+45+50+55+60*0.15 /
(0.7*2+0.5+0.3+ 8*0.15) = **ca 22**

Sugeno: Oppgaven gir singeltons i 10,30 og 50 (erstatter Mamdani-fuzzysettene Low, Medium og High).

Sugeno gir (vekta sum av klippet singeltons) = (0.7*10+0.15*30+0.15*50)/(0.7+0.15+0.15)=**ca 19**
Singeltons under blir altså klippet med til hhv 0.7, 0.15 og 0.15



Begge vurderer altså risken til lav, på grensa til medium.