

Matematiske modeller i kunstig intelligens

Af Thomas Bolander, DTU Compute

Kunstig intelligens er ikke blot én bestemt teori, teknik eller metode, men dækker over en lang række forskellige teknikker til at simulere aspekter af menneskelig kognition på en computer. Forskelligheden i teknikkerne dækker blandt andet over, at de forsøger at efterligne forskellige typer af intelligens, fx sproglig, social eller logisk intelligens. Forskelligheden dækker også over, at vi forsøger at efterligne intelligens på forskellige abstraktionsniveauer gående fra direkte forsøg på at efterligne de atomare neurologiske processer i hjernen helt op til meget abstrakte modeller af vores bevidste, sproglige og logiske tænkning. Fælles for alle teknikkerne er, at det handler om at skabe matematiske modeller af aspekter af kognition, og at få computere til at regne på disse modeller. Desuden handler det om at få kunstig intelligens-systemerne til selv at skabe modeller af deres omverden, som de kan bruge til at ræsonnere om denne omverden. I denne artikel vil vi først give en introduktion til kunstig intelligens og de forskellige hovedparadigmer indenfor området. Dernæst vil vi gå lidt mere i detaljen med, hvordan logiske modeller kan bruges til at skabe ræsonnerende robotter.

Introduktion til kunstig intelligens

“Kunstig intelligens er videnskaben og ingeniørkunsten at lave intelligente maskiner, særligt intelligente computerprogrammer.” Således forsøgte John McCarthy, den kunstige intelligens’ fader, at definere begrebet for 60 år siden. Definitionen er sådan set stadig acceptabel her 60 år senere, men den forklarer bare ikke særligt meget. For vi ved ikke helt præcis hvad intelligens er, og derfor kan vi heller ikke med nogen rimelig præcision sige, hvad det betyder, at en maskine er intelligent. Følgende kan vi dog sige. Kunstig intelligens handler stort set altid om at få computere og robotter til at gøre ting, som det hidtil kun har været mennesker, der har kunnet, fx at spille skak, køre bil, diagnosticere patienter eller føre en samtale.

Når vi gerne vil have computere til at gøre ting, som ellers har været forbeholdt os mennesker, er den naturlige inspirationskilde for programmeringen naturligvis også os mennesker. Vi prøver at forstå hvordan vi selv kan finde ud af at spille skak, køre i bil osv., og så prøver vi at efterligne aspekter af det i en computer. Det kan foregå ved, at vi forsøger at lave simplificerede matematiske modeller af den slags bevidste, logiske ræsonnementer mennesker benytter, når de løser en given opgave som fx at spille skak. Disse modeller bliver så proppet ind i en computer, så den kan regne på dem. Denne tilgang leder typisk til det der kaldes *symbolsk kunstig intelligens*, som vil blive uddybet nedenfor. Alternativt kan man forsøge at lave simplificerede matematiske modeller af de fundamentale neurologiske processer i den menneskelige hjerne og implementere dem i en computer. Det leder typisk til det der kaldes *subsymbolsk kunstig intelligens*. Fælles for dem er, at det handler om at lave matematiske modeller, som kan repræsentere og simulere aspekter af menneskelig kognition, for derigennem at kunne efterligne disse aspekter på en computer.

Det er vigtigt at understrege, at de matematiske modeller af menneskelig kognition vi har i dag, allesammen er meget simplificerede i forhold til de processer, der foregår i den menneskelige hjerne. Selvom moderne computere kan blive verdensmestre i skak eller go

eller være bedre end mennesker til at genkende visse typer af mønstre i billeder (fx hudkræft), er de altså meget langt fra at være på et menneskeligt kognitivt niveau. Ofte kan computere dog kompensere for deres manglende kognitive niveau ved den overvældende beregningskraft, de efterhånden er i besiddelse af, og netop dette er en stor del af hemmeligheden bag de seneste års kraftige vækst i kunstig intelligens. Vi har meget gode matematiske modeller for strategisk ræsonnering (fx til skak) og centrale dele af det visuelle perceptionssystem (fx til billedgenkendelse), men vi har indtil videre ret ringe modeller for vores sproglige og sociale intelligens. Det er derfor indtil videre en meget stor udfordring at lave computere og robotter, som kan komme i nærheden af vores niveau indenfor disse områder.



Figur 1. Et kortudsnit og en tilhørende grafrepræsentation.

Symbolsk kunstig intelligens

I den symbolske gren af kunstig intelligens forsøger man direkte at skabe simplificerede modeller af nogen af de højeste niveauer af menneskelig kognition: vores sproglige, bevidste, logiske tænkning. De matematiske modeller man bygger, er i vid udstrækning baseret på diskret matematik, det vil sige områder som kombinatorik, logik, grafteori, spilteori, algoritmik m.m. Som et meget simpelt eksempel på symbolsk kunstig intelligens kan man tænke på rutefindingsalgoritmerne i GPS-systemer og Google Maps. Her er kortet repræsenteret som en graf, hvor kryds er punkter og vejafsnit er kanter, se figur 1. At finde en korteste (eller “bedste”) ”

rute fra eksempelvis Skælskør til Slagelse handler da blot om at finde en korteste (eller “bedste”) sti mellem de to punkter i grafen, som repræsenterer de to byer. Metoden er i princippet så simpel, at man kan diskutere om den fortjener at blive kaldt for kunstig intelligens. Algoritmer til rutefinding er dog alligevel standardpensum i kurser i kunstig intelligens, og algoritmerne bruges ikke kun til rutefinding på kort, men også til rutefinding for computerstyrede karakterer i computerspil, og til mere generel planlægning af robotters handlinger og bevægelser.

Noget af det der er karakteristisk ved symbolsk kunstig intelligens er, at de modeller man bygger, er *eksplicitte*. Hvis man for eksempel skal konstruere en skakcomputer med symbolsk kunstig intelligens, vil man have de mulige tilstande i spillet eksplicit repræsenteret i computeren, og reglerne vil også være eksplicit beskrevet, fx på en logisk form. Det betyder, at det med symbolsk kunstig intelligens er relativt let at få indsigt i, hvordan computeren “tænker”, og det er relativt let at modificere dens adfærd (fx at justere den til at kunne spille andre spil). Intelligente personlige assistenter som Siri på iPhone, Amazon Alexa og Google Home er også baseret på symbolsk kunstig intelligens, på nær talegenkendelsen (tale-til-tekst konvertering), som gøres subsymbolsk (se næste afsnit). Symbolsk kunstig intelligens har typisk den fordel, at den kan gøres robust og forudsigelig, og man vil også kunne give garantier for dens adfærd. Man kan fx matematisk bevise, om en given rutefindingsalgoritme altid finder den korteste rute eller ej. Ulempen ved symbolsk kunstig intelligens er så til gengæld, at de systemer man bygger typisk har nøje afgrænsede evner, og typisk ikke lærer af deres erfaringer.

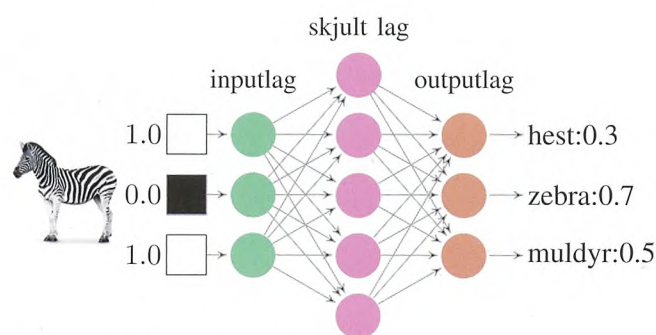
Subsymbolsk kunstig intelligens

Hvor den symbolske kunstige intelligens direkte forsøger at efterligne de højeste niveauer af menneskelig kognition, har subsymbolsk kunstig intelligens den diametralt modsatte tilgang. Her prøver man i stedet at bygge intelligens op nedefra igennem modeller af de mest fundamentale processer i den menneskelige hjerne. Det klassiske eksempel er kunstige neurale netværk, som er en simplificeret model af de atomare, neurologiske processer i den menneskelige hjerne. Den matematik man bruger til subsymbolsk kunstig intelligens er typisk lineær algebra og differentiaalligninger.

Som konsekvens af den tilgang til modellering, som subsymbolsk kunstig intelligens har, er denne form for kunstig intelligens typisk bedst til at efterligne de mest grundlæggende niveauer af menneskelig kognition som vores perception. Kunstige neurale netværk er efterhånden ret effektive til at efterligne en lang række aspekter af menneskelig perception som genkendelse af billeder og lyd. Det kan være sådan noget som at kunne kende forskel på billeder af hunde og katte eller kunne høre forskel på om nogen siger “tirsdag” eller “torsdag”. Kunstige neurale netværk bliver ikke eksplicit programmeret med forskellen mellem hunde og katte, men bliver trænet på en masse billeder af hunde og katte, hvor de får at vide hvad der er hvad.

Gennem en masse eksempler på objekter indenfor de forskellige kategorier lærer det (kunstige) neurale netværk efterhånden selv at kunne foretage kategoriseringer. Det er ret afgørende for kunstig intelligens, at vi har subsymbolske metoder til at kunne genkende mønstre og lave kategoriseringer af objekter. Det ville være temmeligt meget op ad bakke at skulle bruge symbolske metoder til at kende forskel på hunde og katte, da det ville kræve at vi programmerede computeren med en eksplicit, symbolsk karakterisering af forskellen mellem de to dyr. Omvendt er vi også stadig afhængige af symbolske teknikker til at kunne simulere kognitive processer på højere niveau, som fx logisk ræsonnering eller planlægning.

Fordi kunstige neurale netværk lærer igennem træning, hører de under den del af kunstig intelligens der kaldes maskinlæring (machine learning). Der findes dog også maskinlæring indenfor den symbolske gren, det kan fx være systemer der forsøger at lære eksplicitte, logiske repræsentationer af reglerne i skak eller whist ved at observere andre spille disse spil.



Figur 2. Et kunstigt neuralt netværk.

Et kunstigt neuralt netværk er opbygget af neuroner (punkter) og neuronforbindelser (kanter), se figur 2. Neuronerne er placeret i lag, og hver neuron modtager et signal fra hver af neuroner i det foregående lag (i et fuldt forbundet netværk). Neuronen omsætter disse inputsignaler til et outputsignal, som den sender videre til det næste lag. På den måde fungerer hele det neurale netværk som en signalprocessor, der omsætter et input-signal til et outputsignal. Inputsignalet kunne fx være pixels fra et billede af et objekt, man ønsker klassificeret, og outputsignalet kunne så være klassifikationen. Figuren viser et simplificeret eksempel, hvor vi med et neuralt netværk ønsker at kunne skelne mellem billeder af zebraer, heste og muldyr. Der er derfor 3 neuroner i outputlaget, én for hver af de tre dyr. Når et billede skal klassificeres, sender man det således ind i inputlaget (med én inputneuron per pixel), hvorefter det flyder igennem netværket, og klassifikationen er så givet ved den outputneuron, som producerer det kraftigste signal (zebra-neuronen i figuren). Træningen af neurale netværk foregår ved at justere vægtene på neuronforbindelserne, som bestemmer hvor kraftigt de forskellige inputsignaler bliver forstærket. Hvis det neurale netværk under træningen fejlklassificerer en zebra som en hest, vil det automatisk justere vægtene, så outputsignalet på tilsvarende billeder i fremtiden vil komme tættere på at give den rigtige klassificering.

Sammenligning af symbolske og subsymbolske metoder

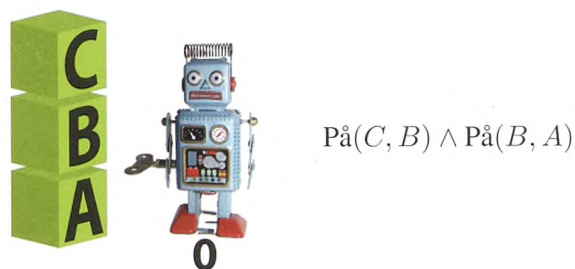
Vi nævnte tidligere, at de matematiske modeller benyttet i symbolsk kunstig intelligens er *eksplicitte*: vi opbygger eksplicitte, symbolske modeller af det domæne, vi ønsker at modellere. I subsymbolsk kunstig intelligens er de *implicitte*. Vi kan ikke umiddelbart inspicere et neuralt netværk, som kan kende forskel på heste, zebraer og muldyr, og så finde ud af præcist hvad klassificeringen er baseret på. Klassificeringen er opnået gennem træning og er udelukkende repræsenteret implicit via de trænedede neuronvægte. Det har følgende konsekvenser. Modsat symbolsk kunstig intelligens kan vi ikke i subsymbolsk kunstig intelligens give garantier for adfærd, vi kan fx ikke bevise om et neuralt netværk altid vil kategorisere zebraer korrekt. Neurale netværk kan aldrig være 100% fejlfri og robuste. Symbolske og subsymbolske teknikker har dermed hver deres styrker og svagheder, ligesom de også har hver deres anvendelsesområder.

Man kan give følgende analogi omkring forskellen mellem symbolsk og subsymbolsk kunstig intelligens. Antag at du ønsker at forudsige hvor langt en kanonkugle vil flyve afhængigt af parametre som vinkel på kanonrøret, størrelse af ladning og vindforhold. En mulig metode til at forsøge at udregne rækkevidden af kanonkuglen kan da være at opstille en bevægelsesligning baseret på mekanikkens love, og så regne på den ligning. Det vil være en eksplicit modellering i stil med symbolsk kunstig intelligens: Vi har en eksplicit, symbolsk model for det fænomen vi studerer, og når vi skal lave forudsigelser, så regner vi på denne eksplicitte model. En anden mulig metode kunne være at observere et stort antal kanonaffyringer under forskellige forhold, og så prøve at blive i stand til at lave præcise forudsigelser uden først at skulle forsøge at opstille eller forstå mekanikkens love på symbolsk form. Det svarer lidt til, hvordan vi som børn lærer at kaste en bold præcist. For at kunne ramme præcist, bliver vi nødt til at have en model af, hvordan bolden bevæger sig (i hvert fald af hvor langt den vil bevæge sig), men det betyder jo ikke, at vi løser symbolske bevægelsesligninger i hovedet, det betyder blot, at vi har fået opbygget en tilstrækkeligt præcis *implicit* model. Det er det samme, som sker i neurale netværk. Et neuralt netværk vil kunne trænes op til at forudsige rækkevidden af en kanonkugle, men vi vil ikke kunne inspicere neuronvægtene og deraf udlede noget som ligner bevægelsesligninger.

Symbolsk modellering af omverden

Ovenfor har vi snakket om kunstig intelligens som et modelleringsproblem: Vi mennesker laver matematiske modeller af aspekter af kognition eller ræsonnering, som vi implementerer på en computer. Men de implementerede kunstig intelligens-systemer er også systemer, som selv løser et modelleringsproblem: De skal skabe modeller af deres omverden, som de kan bruge til deres beslutninger, forudsigelser og klassifikationer, hvad enten det handler om at spille skak eller kende forskel på hunde og katte. Kunstig intelligens er således ofte en slags metamodelering, fordi vi ønsker at skabe

modeller, som selv er i stand til at skabe modeller af deres omverden. Neurale netværk er et godt eksempel på dette, da et neuralt netværk gennem træning på meget direkte vis skaber en (implicit) model, som kan bruges til forudsigelser, beslutninger eller klassifikation.



Figur 3. En simpel verden og den tilhørende logiske model.

I det følgende vil vi fokusere på, hvordan symbolsk kunstig intelligens modellerer sin omverden. Der er mange forskellige tilgange, men det mest udbredte er at modellere omverden igennem logik. Figur 3 giver et simpelt eksempel. Den blå robot (robot 0) observerer en stak med tre kasser markeret med C , B og A , hvor C er øverst og A nederst. En simpel logisk model af denne verden er givet ved formelen vist til højre på figuren, som udtrykker at " C er ovenpå B og B er ovenpå A ". En symbolsk programmeret robot vil kunne bruge en sådan logisk formel til at beskrive sin omverden. Hvis robotten kan flytte rundt på kasserne, skal den også forstå hvordan verden, og dermed den logiske formel, ændrer sig når den gør det. Det kan også gøres logisk via det der hedder handlingsskemaer. Følgende handlingsskema beskriver logisk hvad det vil sige at tage et objekt x og putte det ovenpå objektet y :

ACTION: Put(x, y)
PRECONDITION: På(x, z) \wedge $\neg\exists v$ På(v, x) \wedge $\neg\exists w$ På(w, y)
EFFECT: \neg På(x, z) \wedge På(x, y)

Den logiske formel PRECONDITION beskriver betingelsen for at man kan putte x ovenpå y :

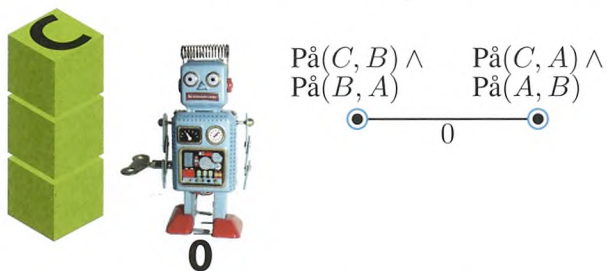
- På(x, z): x skal allerede være ovenpå et objekt z ,
- $\neg\exists v$ På(v, x): der må ikke være noget objekt ovenpå x og
- $\neg\exists w$ På(w, y): der må heller ikke være et objekt ovenpå y .

Betingelsen kan eksempelvis *ikke* være opfyldt, hvis x er kasse B ovenfor, for kasse C ligger ovenpå kasse B . Den logiske formel EFFECT beskriver, hvordan verden ændrer sig, når handlingen bliver succesfuldt udført: x er ikke længere ovenpå z , men ovenpå y i stedet. Hvis vi anvender handlingen Put($C, gulv$) på situationen vist i figur 3, vil resultatet blive, at På(C, B) ikke længere er sand, men at På($C, gulv$) i stedet bliver sand. Den resulterende formel bliver altså:

$$\text{På}(C, \text{gulv}) \wedge \text{På}(B, A)$$

Dette er naturligvis som ønsket: Hvis robotten flytter kasse C ned på gulvet, så vil C nu være på gulvet, mens B stadig er på A .

Ideen er her at robotten både har en logisk model af sin omverden og logiske modeller af hvordan dens handlinger modificerer omverden, og på den måde kan den let forudsige effekten af forskellige mulige handlinger. Det muliggør, at den kan lægge planer. Vi kan eksempelvis bede robotten om at lægge kasserne i alfabetisk rækkefølge med A øverst, og det problem vil den let kunne løse ved at regne på sine mulige handlinger, og hvordan de modificerer verden. Faktisk kan man til denne del af problemet bruge de samme typer af algoritmer, som man bruger til rutefinding, som var det første eksempel på kunstig intelligens vi så på.



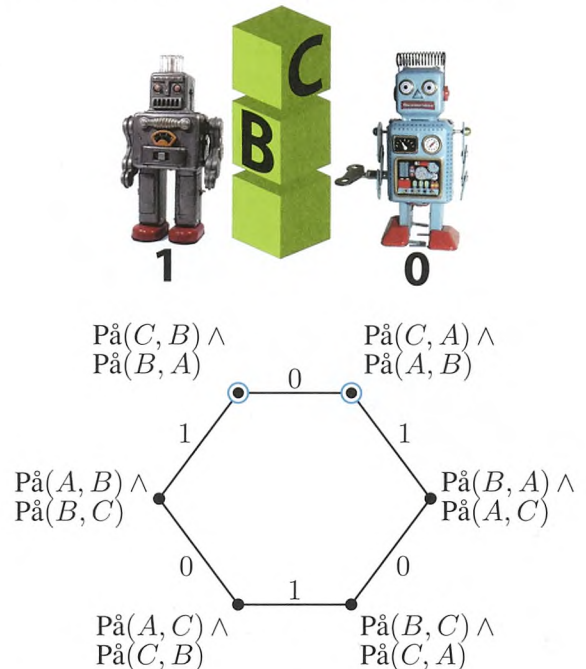
Figur 4. En verden uden fuld observerbarhed.

Robotter som skaber modeller af andre robotter

Det er ikke altid at en robot kan lave en præcis verdensbeskrivelse, der kan eksempelvis være ting, som den ikke kan se. Antag eksempelvis at kasserne i stedet er placeret med bogstaverne vendt opad, så robotten kun kan se bogstavet på den øverste kasse, som vist på figur 4. I dette tilfælde er robotten usikker på, om A er ovenpå B eller omvendt. Denne usikkerhed kan vi repræsentere ved at lave en model, som indeholder begge muligheder, som vist til højre i figur 4. I stedet for en enkel logisk formel er modellen nu en graf, hvor punkterne repræsenterer forskellige mulige verdener, og disse mulige verdener er hver især beskrevet som en logisk formel. Kanten i grafen mellem de to mulige verdener repræsenterer, at robotten ikke kan skelne disse to verdener: den ved ikke om det er B der er på A (den venstre verden i figur 4) eller A der er på B (den højre verden).

Ideen med at repræsentere sine omgivelser gennem mulige verdener bliver endnu mere afgørende, hvis robotten ikke længere er alene i verden. Betragt situationen i figur 5, hvor der nu er en ekstra robot (robot 1). Robot 1 kan kun se bogstavet på den midterste kasse, mens robot 0 kun kan se bogstavet på den øverste kasse. Grafen under robotterne viser hvordan denne situation kan modelleres af robot 0. De to punkter med en ekstra cirkel omkring er de samme som i figur 4. De repræsenterer de to verdener, som robot 0 opfatter som mulige. Men der er nu også en række nye punkter, og der er nu også både 0-kanter og 1-kanter i grafen. En 1-kant mellem to mulige verdener betyder, at robot 1 ikke kan skelne de to verdener. Det overlades til den interesserede læser at tjekke, at modellen faktisk repræsenterer den viste situation, og de usikkerheder

som de to robotter har med hensyn til, hvordan den virkelige verden ser ud. Pointen med modellen er, at det ikke længere kun er en model af, hvad robot 0 ved om kassernes konfiguration, men også en model af hvad robot 0 ved om robot 1. Ethvert par af mulige verdener, som kun afviger fra hinanden ved at der er byttet om på den øverste og nederste kasse, er altid forbundne af en 1-kant, og det repræsenterer netop det faktum, at robot 1 kun kan se bogstavet på den midterste kasse.

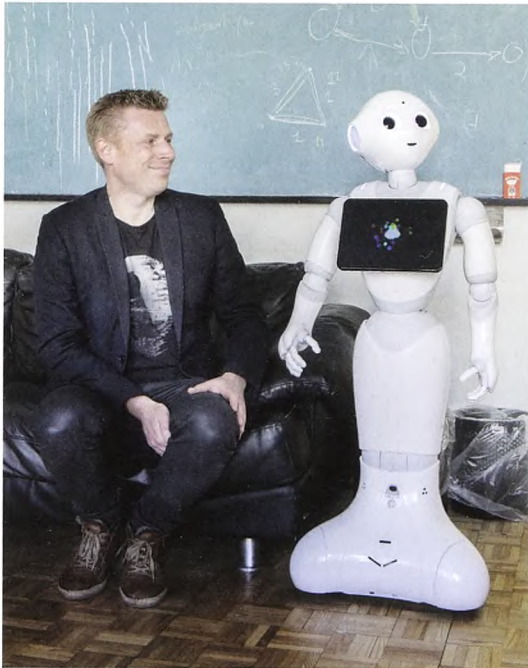


Figur 5. En verden med flere robotter.

Beder vi nu robot 0 om at placere kasserne i alfabetisk rækkefølge med A øverst, vil den ikke kunne løse opgaven alene, for den ved ikke om A er ovenpå B eller omvendt. Men den ved faktisk, at robot 1 ved om A eller B er øverst, for den ved, at robot 1 kan se bogstavet på den midterste kasse. En mulig måde for robot 0 at løse problemet på vil derfor være først at spørge robot 1, om bogstavet på den midterste kasse er A , og afhængigt af om svaret er ja eller nej, vil den så vide hvad den skal gøre. Ved at give robotter mulighed for at skabe og ræsonnere med denne slags modeller, kan vi således også få dem til at løse problemer, hvor de på denne måde bliver nødt til at sætte sig i andres sted og interagere med andre for at løse deres problemer.

Perspektiver og anvendelser

Modeller af ovenstående type muliggør, at kunstig intelligens-systemer kan sætte sig i andres sted og se verden fra andres perspektiv. Det er den evne som i kognitiv psykologi kaldes Theory of Mind. Der er solid evidens for at Theory of Mind er afgørende for menneskers sociale intelligens. Vores evne til at sætte os i hinandens sted er afgørende for vores evne til at samarbejde, være hjælpsomme og endda blot at forstå hvad hinanden mener, når vi siger noget. Mennesker formulerer sig generelt meget upræcist og flertydigt, noget man typisk først lægger mærke til, når man begynder at forsøge at konstruere kunstig intelligens-



Figur 6. Thomas Bolander og Pepper-robotten R2DTU.

systemer, som kan kommunikere med mennesker. Til trods for dette har vi en ret veludviklet evne til at forstå hinanden, og det skyldes i høj grad at vi kan sætte os i hinandens sted og forstå intentionen bag det sagte. Modeller som ovenstående er kun de første spæde skridt i retningen mod kunstig intelligens med bedre sociale evner og en generel evne til at sætte sig i andres

sted, men der er ingen tvivl om, at det er den rigtige retning at gå, hvis vi ønsker en fleksibel, letflydende og robust interaktion med fremtiden robotter, computere og smartphones.

Modeller som ovenstående er specifikt blevet benyttet til at vise, hvordan robotter kan bestå såkaldte *false-belief tasks*, en type af tests som oprindeligt er udviklet til at vurdere børns evne til at sætte sig i andres sted. I skrivende stund eksperimenterer vi på DTU med at få robotter af typen Pepper (figur 6) til at bestå sådanne false-belief tasks samt andre typer af opgaver, som kræver social perspektivtagning og sætten sig i andres sted. Vi arbejder også med scenarier indenfor eksempelvis hospitalsrobotter og husholdningsrobotter, hvor der er mange eksempler på, at det kræver en evne til at modellere andres intentioner, tanker og viden for at kunne agere på en socialt acceptabel måde.



Thomas Bolander er lektor ved DTU Compute. Hans forskning er centreret om kunstig intelligens med særligt fokus på social intelligens. Thomas Bolander er et af kernemedlemmerne af SIRI-kommissionen og en førende dansk foredragsholder indenfor kunstig intelligens.

Foreningsnyt – kommende foredrag

Dato	Tid	Foredragstitel	Foredragsholder	Forening
Juni				
29 eller 30/6	19.00	Asteroid Day (sted og dato er ikke fastsat)	Check www.naturvidenskab.net	SNU
September				
3/9	19.15	Opdagelsesrejse på Mars	Jens Frydenvang	AS (Kbh)
10/9	19.00	Opdagelsesrejse på Mars	Jens Frydenvang	AS (Aarh)
10/9	19.30	Oversvømmet – igen! Udfordringer med for meget vand	Carlo Sørensen	SNU
24/9	19.15	Exoplaneter og stjernesvingninger	Rasmus Handberg	AS (Kbh)
Oktober				
1/10	19.00	Exoplaneter og stjernesvingninger	Rasmus Handberg	AS (Aarh)
1/10	19.30	Store jordskælv og hvad deraf følger	Trine Dahl-Jensen	SNU
15/10	19.15	Tyngdebølger, kilonovaer og oprindelsen af guld	Jonatan Selsing	AS (Kbh)
22/10	19.00	Tyngdebølger, kilonovaer og oprindelsen af guld	Jonatan Selsing	AS (Aarh)
22/10	19.30	Vulkanisme: årsag og virkning	Paul Martin Holm	SNU
November				
5/11	19.15	Udforskning af det højenergetiske univers	Desiree Della Monica Ferreira	AS (Kbh)
12/11	19.00	Udforskning af det højenergetiske univers	Desiree Della Monica Ferreira	AS (Aarh)
12/11	19.30	Pandemier – historiens store dræbere	Nils Strandberg Pedersen	SNU
26/11	19.15	Verdens mest avancerede robotteleskop	Mads Fredslund Andersen	AS (Kbh)
December				
3/12	19.00	Verdens mest avancerede robotteleskop	Mads Fredslund Andersen	AS (Aarh)
10/12	19.30	Når Jorden står for skud – asteroidenedslag og livets udvikling på Jorden	Henning Haack	SNU

AS (Kbh): Astron. Selskab (Kbh), Aud. 2, H.C. Ørsted Institut, Universitetsparken 5, 2100 København Ø (astronomisk.dk).
AS (Aarh): Astron. Selskab (Aarh), Matematisk Institut, AU, Ny Munkegade 118, Bygn. 1530, Aud.F/G122, 8000 Aarhus C.
SNU: Aud. 1, H.C. Ørsted Institut, Universitetsparken 5, 2100 København Ø (naturvidenskab.net, facebook.com/SNU1824).
 Bemærk, at tid og sted for Asteroid Day ikke er fastsat.