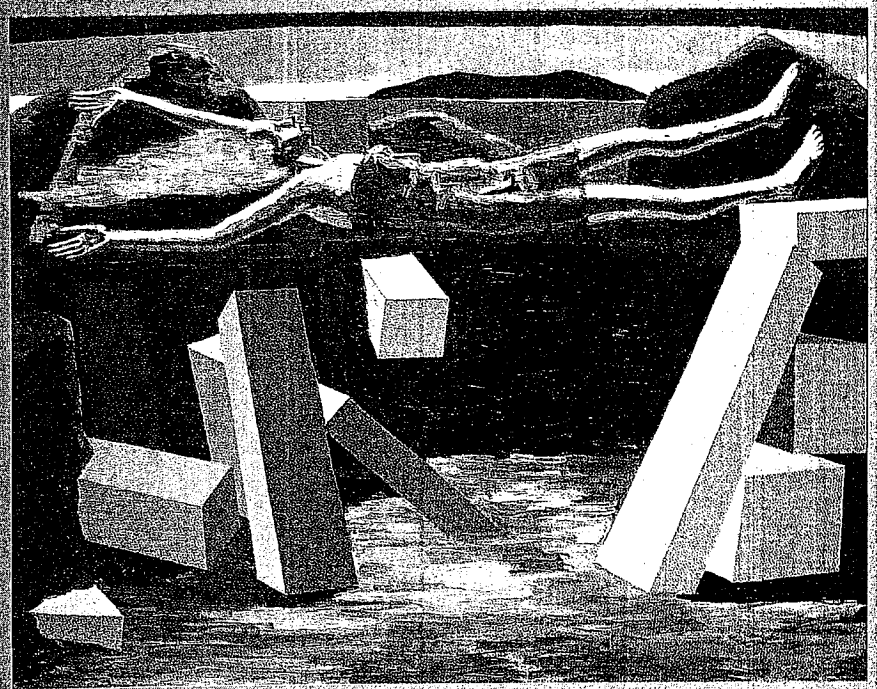
Redaktion — Niels Ole Finnemann og Frederik Stjernfelt



KOGNITION *og* SPROG

KULTURSTUDIER · 14

Udgivet af Center for Kulturforskning
Aarhus Universitet

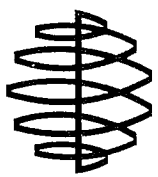
ISSN 0904 0013
ISBN 87-7288-364-2

Sat med New Century
Tryk: Specialtrykkeriet-Viborg a/s
Lay-out: Inger Hein

ISSN 0904-0013
ISBN 87-7288-364-2

Omslag: Vanitasbilleder.
Diana på kirkegården - Diana svævende af Inge Ellegaard.

Aarhus Universitetsforlag
Aarhus Universitet
8000 Århus C
Tlf. 86 19 70 33



KULTURSTUDIER 14

Skriftserien KULTURSTUDIER udgives af
Center for Kulturforskning v/Aarhus Universitet,
Finlandsgade 26, 8200 Århus N.
Redaktion: Jørgen Østergård Andersen, Frands Mortensen
og Anders Troelsen.

Indhold

Får man virkelig noget for intet med kunstige neurale netværker?

I denne artikel vil jeg beskrive et lingvistisk problem man støder på i forbindelse med programmer der kan lave maskinoversættelse. Det drejer sig om hvordan man entydiggør de polyseme (mangetydige) ord som findes i enhver tekst. Først vil jeg vise hvordan man kan løse problemet med et traditionelt serielt program (kunstig intelligens), det virker, men er meget langsommeligt og klodset. Dernæst vil jeg vise en anden løsning på problemet, en med parallelle kunstige neurale netværker; denne løsning er hurtig og elegant. På dette grundlag vil jeg beskrive hvad det er man opnår med kunstige neurale netværker sammenlignet med klassisk kunstig intelligens. Er det rigtigt at man får noget for intet med kunstige neurale netværker?

Det lingvistiske problem

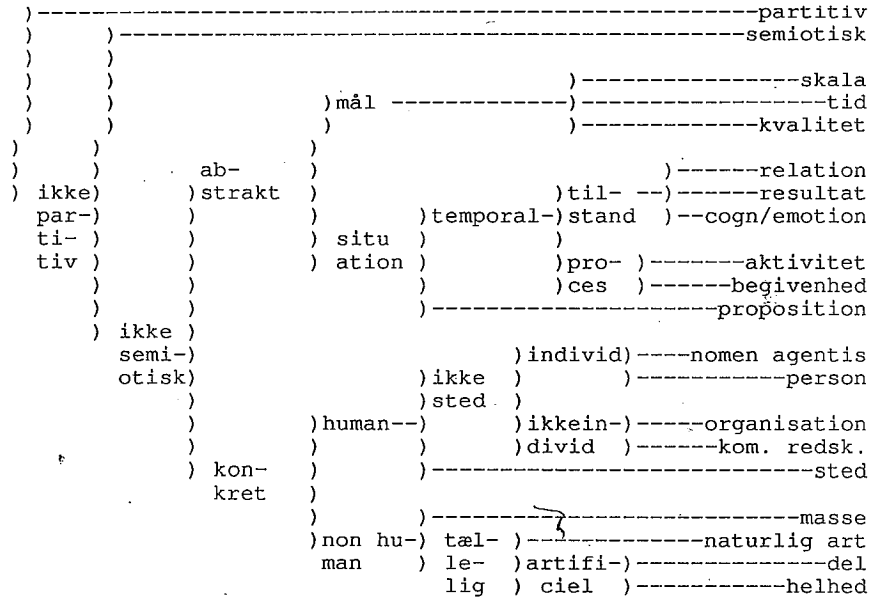
Det lingvistiske problem kan illustreres med følgende eksempel:

Lenin boede i 1897 i hemmelighed i København i 4 dage

Problemet er at ordet *i* betyder noget forskelligt hver af de 4 gange det forekommer i sætningen, nemlig 1: 'på tidspunktet', 2: 'under tilstanden', 3: 'på stedet' og 4: 'med en varighed af'. I selve lingvistikken er problemet om ordenes polysemi temmelig overset; i leksikografien løser man problemet traditionelt ved at sige at et ord har flere forskellige betydninger som fx angivet ovenfor med *i* (i ODS er der anført 16 betydninger). Selv hælder jeg til den lingvistiske teori at ord ikke har præcise betydninger i sig selv (dvs. løsrevet fra den sammenhæng de bruges i i en ytring), men at et ord først får præcis betydning når de ytres i en bestemt kontekst, og at konteksten så også er en del af den betydning som ordet tillægges.

I transformationsgrammatikken beskrives betydningsvarianterne af et ord ved deres forskellige selektionsrestriktioner; man kan fx sige at ordet *i* betydningsvariant 1: 'på tidspunktet' af sin styrelse (det led som står efter *i*) kræver at det hører til kategorien dato/tid, i betydningsvariant 2: 'under tilstanden' af sin styrelse kræver at den hører til kategorien tilstand osv. Der er imidlertid også den lingvistiske regel at ordet *i* også kan være en slags afledningsendelse, fx i forhold til verbet *bo*; *i*'et angiver hvilket ord der er middelbart objekt (genstandsled) for verbet. I nogle grammatiske beskrivelser vil man analysere *bo i København* som verbal (*bo*) + præpositionsforbindelse (*i København*), men der er meget der taler for at en anden analyse er bedre: frasen består af et diskontinuert præpositionsverbal (*bo_i*) + objekt (*København*). Det er så dette præpositionsverbal (*bo_i*) der af sit objekt kræver at det hører til kategorien lokalitet. Læg mærke til at præpositionen i præpositionsverbalet ikke nødvendigvis er *i*, det kan være alle de præpositioner der kan angive steder, fx *bo_på, bo_ved, bo_hos*, men ikke dem der ikke kan angive steder fx ikke **bo_uden*, eller **bo_med*.

Som man kan regne ud forudsætter denne selektionsregel at alle substantiver, nemlig alle de mulige styrelser i præpositionsforbindelser, og alle de mulige objekter, beskrives som hørende til en af de kategorier som præpositioner og verber selekterer for. Systemet af kategorier som substantiver kan tilhøre, kan jeg foreslå er følgende:



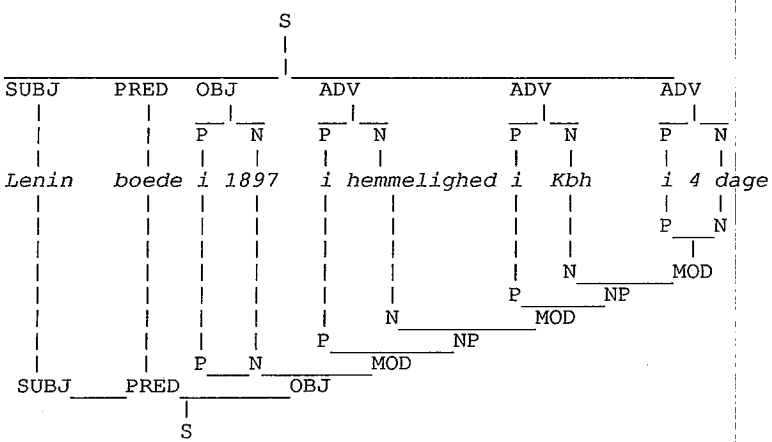
Eksempler: partitiv: *sektor, side, halvdel*, semiotisk: *afsnit, forslag, aftale*, skala: *meter, decibel, grad, minut, dag*, tid: *efterkrigstiden, fremtiden, 1987*, kvalitet: *identitet, størrelse, længde*, relation: *afhængighed, faktor, position, hemmelighed*, resultat: *produktion_2, undtagelse, investering_2*, cogn/emotion: *interesse, frygt, glæde*, aktivitet: *databehandling, anvendelse, produktion_1*, begivenhed: *revolution, investering_1, udforskning*, proposition: *fordel, mulighed, problem*, nomina agentis: *fabrikant af, tilskuer til, herre over*, organisation: *hjemmemarked, industri, regering*, kommunikationsredskab: *persondatamat, videobåndoptager, radio*, sted: *europa, københavn, ildlinjen*, masse: *vand, luft, sand*, naturlig art: *blomst, træ, sten*, del: *kredsløb, svinghjul, taster*, helhed: *dataanlæg, elektronik, informationsteknologi.*

Der er selvfølgelig en række kriterier for hvorledes man kan se at et ord hører til en kategori. Det er jo nemlig en distributionel egenskab ved ordet at det hører til en kategori, og denne distribution kan findes ved en række tests af typen : Giver det mening at sige: *Jeg så **** forsvinde i det fjerne.* Hvis det giver mening at sætte ordet ind på stjernernes plads hører ordet til overkategorien konkret, hvis ikke, til overkategorien abstrakt. På denne måde kan man for hvert ord i hver af dets betydningsvarianter afgøre hvilken kategori det hører til.

Parseren

Problemet om ordenes betydningsvarianter opstår først i den elektroniske tekstanalyse når man har løst problemet om hvorledes maskinen kan parse ("læse" og "analysere") sætninger som input og producere en repræsentation af sætningens grammatiske struktur som output. Programmet der gør det, består af en grammatik, et leksikon, en parser. Grammatikken består af en række regler for hvilke led en sætning består af, hvilke led disse led så består af, og hvilke typer af ord (ordklasser) der indgår i disse led, samt for rækkefølgen af disse ord og led. Leksikonet består af lister over alle de ord der findes i sproget, med angivelse af deres ordklasse (dvs. deres muligheder for at blive brugt som materiale i reglerne). Parseren består af regler for hvorledes maskinen med grammatikkens og leksikonets regler tager sætningen som input og producerer analyse-træer over sætningens mulige strukturer som output.

Med grammatik, leksikon og parser får man, med programmer som jeg ikke skal omtale nærmere her, lavet et vist antal analysetræer som hver angiver de mulige læsninger som sætningen efter grammatikken og leksikonet kan have. De fleste sætninger giver flere mulige syntaktiske analyser. Af denne sætning er der mulighed for et helt fladt træ, et helt dybt træ, og 6 blandingstræer ind i mellem.

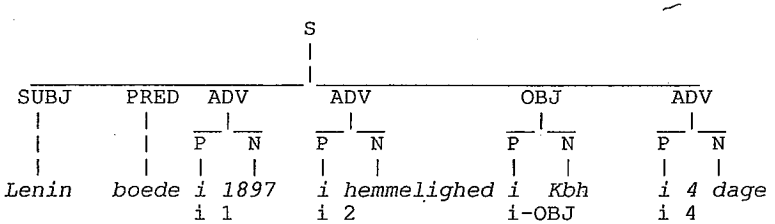


Selektionsmekanismen i kunstig intelligens-programmer

Ovenstående analyse forudsætter at der kun er ét ord der hedder *i*. Men i leksikonet skal der ikke blot stå ét ord der hedder *i*, men lige så mange som der er betydningsvarianter, de kan så hedde: *i_1, i_2, i_3, i_4* osv. Maskinen vil nu lave et træ med *i_1* i den første position (*i 1897*), *i_1* i anden position (*i hemmelighed*), i_1 i tredje position (*i København*), et træ med *i_2* i den første position, *i_1* i anden position, *i_1* i tredje position osv., et træ med *i_3* i den første position osv. Foruden de 8 syntaktisk mulige analysetræer, vil maskinen således lave (i vores eksempel kun) 4x4x4x4 træer, eller 256 x 8 analysetræer. Når maskinen således har produceret alle de mulige læsninger af sætningen er der brug for en mekanisme der ved hjælp af selektionsreglerne får sorteret den rigtige læsning frem.

Denne selektionsmekanisme kan i et serielt kunstig intelligensprogram være udformet på den måde at der i ordbogen under *i_1* står at det selekterer kategorien dato/tid, under *i_2* at det selekterer tilstand, osv. I ordbogen står der så også for alle nominer (substantiver) hvilken kategori de hører til, altså fx at ordet *afsnit* hører til kategorien semiotisk. Reglen består i at maskinen går ind for hvert af de producerede (her 2048) analysetræer og tjekker om nominerne matcher med hvad præpositionerne kræver efter selektionsreglerne. Hvis de ikke matcher, hvis fx *i_1* kræver at nominet er dato/tid, men substantivet *hemmelighed* hører til kategorien relation, så bliver det analyse-træ hvori denne struktur forekommer, slettet som fejlagtig.

I dette eksempel vil reglerne på denne måde slette 255 af de 256 producerede analysetræer fordi der et eller andet sted i træet ikke er match mellem præpositionensselektion og nominet. Det samme gælder i princippet for de 8 forskellige syntaktiske træer, men argumentationen herfor er ikke helt simpel, og derfor udelader jeg det her. Det eneste træ der vil være tilbage er dette:



Som man kan se af denne skitse, kan man godt få programmet til at virke således at maskinen finder frem til den rigtige analyse af sætningen ved hjælp af grammatikken, et leksikon med kategoriseringer af substantiver og selektionsrestriktioner på hver variant af alle verber og præpositioner, en parser og en selektionsmekanisme - men det tager lang tid, det er besværligt og uelegant, og det er en meget 'skør' regel, forstået således at det ofte vil kikse, nemlig i de tilfælde hvor selektionsreglen faktisk er overtrådt fordi udtrykket er mere eller mindre metaforisk eller metonymisk, fx *han boede midt i sine minder*; *minder* er jo ikke et sted, men ikke desto mindre er det selekteret af *bo_i*.

Også dette problem kan løses inden for rammerne af klassisk kunstig intelligens, nemlig ved præferenceregler der vælger den mindst ringe blandt de producerede analysetræer i stedet for at slette alle de ikke korrekte. Men det betyder kun en endnu langsommere maskine som slet ikke kan konkurrere med den menneskelige hjerne i løsningen af dette problem; maskinen skal snarere bruger timer end minutter på at løse problemet - på trods af at den jo på det fysiske niveau fungerer meget hurtigere end hjernen gør på det fysiske niveau.

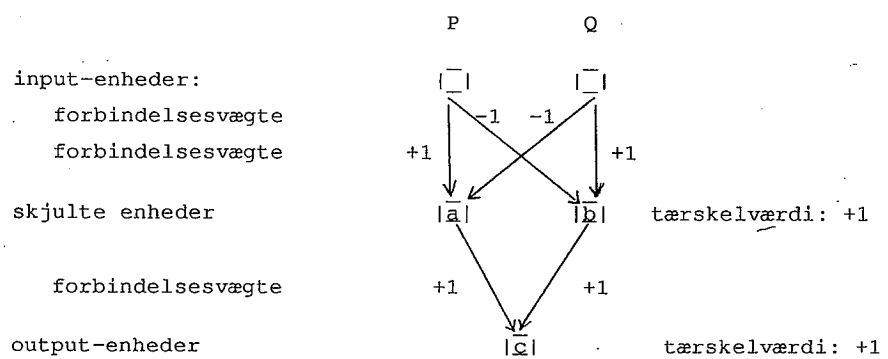
Den måde et kunstig intelligens-program løser dette problem på kan umuligt afspejle hvad der sker inde i hovedet på sprogbrugeren der jo uden betænkning i første omgang griber den rigtige læsning af *i* hver gang det forekommer. Der er ingen der får antydningen af nogen af de forkerte betydninger gennem hovedet ved læsningen af sætningen. Hvem har fx ved læsningen af sætningen tænkt: 'Lenin boede på (nordsiden af) 1897, på tidspunktet hemmelighed, med en varighed af København, under tilstanden 4 dage'? Det kunstige intelligens-program bærer således ikke med rette sit navn; det løser ikke problemet på samme intelligente måde som mennesket gør det, men kan med besvær og stort tidsforbrug nå det samme resultat.

Det kunstige neurale netværk

Et kunstigt neuralt netværk er et program der simulerer at maskinen laver parallelle processer i modsætning til den virkelige maskine hvor processerne er serielle, men meget hurtige. Derfor kan

man kalde denne type programmer PDP, parallelt distribuerede processer. Man forestiller sig at programmet simulerer processerne som de foregår i hjernen, hvor elektriske impulser sendes fra en hjernecelle til flere andre, og fra disse til flere andre parallelt. Derfor kalder man så også hver enkelt procesenhed for kunstige neuroner. Det program som her skal demonstreres, er bygget op med en række enheder der modtager impulser udefra (fra programmøren), en række enheder der leverer maskinens output, og imellem dem, en række enheder som kaldes skjulte fordi deres tilstand ikke umiddelbart kan aflæses af programmøren. Mellem disse tre lag af procesenheder er der så forbindelser (*connections*, hvorfor den programmeringsform også kaldes *konnektionisme*). Hver procesenhed i maskinen fungerer således som man forestiller sig at hjernecellerne, neuronerne, fungerer: de får en impuls som input, og de leverer en række impulser til andre procesenheder hvis en tærskelværdi for inputniveauet er overskredet.

Transmissionen til de andre enheder forgår gennem forbindelserne som hver er udstyret med en bestemt vægt (eller modstand), således at inputtet fra en enhed til hver af de næste ikke bliver den samme, men er afhængig af den vægt som er blevet tilskrevet den specielle forbindelse. Hele systemet i et såkaldt feed forward (modsat feed back) netværk kan se ud som i følgende netværk som kan processere det eksklusive *eller* (enten P, eller Q, men hverken P og Q eller ingen af dem):



Man ser at maskinen ved input P vil sende impulsen +1 til den skjulte enhed a, som så vil sende impulsen videre til c, som vil sende en impuls videre; det vil vi tolke som: hele input-udtrykket er sandt. Det samme gælder for et input der hedder Q. Hvis inputtet er hverken P eller Q, vil ingen af de skjulte enheder blive aktiveret, og derfor heller ikke output-enheden c; og det opfatter vi som: hele udtrykket er falsk. Det interessante er nu at hvis inputtet er P og Q, vil både enhed a og enhed b modtage både impulsen +1 og -1. Summen af dem overstiger ikke tærskelværdien +1, og ingen af de skjulte enheder sender derfor impulsen videre. Sandhedsværdien af hele udtrykket er altså falsk.

I dette kunstige neurale netværk har jeg, som en gud der skaber verden, været inde og justeret vægtene på forbindelserne således at hele systemet virker som en enten-eller-maskine. Bliver antallet af enheder i hvert af de tre lag større, bliver det imidlertid umuligt at overskue hvorledes vægtene skal stilles på hver af de tusindvis af forbindelser der vil blive. Derfor har man fundet på et princip efter hvilket man kan træne netværket således at det på grundlag af en række inputeksempler med tilsvarende autoriserede outputværdier (en facitliste) selv justerer vægtene til værdier der producerer det ønskede resultat. Man tager altså enten-eller-maskinen med dens 5 enheder, men stiller alle de 6 vægte tilfældigt. Man giver den så følgende 4 træningseksempler:

input	ønsket output
P og Q	falsk
- P og Q	sand
P og - Q	sand
- P og - Q	falsk

Med de tilfældigt stillede vægte vil maskinen selvfølgelig ikke producere de ønskede output. Træningsprincippet består nu i at den fejl der findes i outputværdien føres tilbage til hver af de brugte vægte således at værdien af vægten tilnærmes til den ønskede værdi med en vis procentdel. Man kalder det tilbageavling (backpropagation) af ændringen. Efter et vist antal træningsrunder med de 4 mulige eksempler, vil alle vægtene være justeret så de danner et eller andet mønster der giver det rigtige output. Man kan her lægge mærke til at der er uendeligt mange mønstre der giver alle de rigtige resultater

og kun dem, og at mønstret ikke bliver det samme ved to forskellige træningsomgange.

Kategorierne i præpositionsnetværket

Jeg har konstrueret et kunstigt neuralt netværk der kan afgøre hvilken variant af præpositionen *i* der er på spil i en given sætning. Det er således indrettet at der er 80 inputenheder, 70 skjulte enheder og 9 output enheder. Inputenhederne angiver kategorien for hver de 4 nærmeste ord til venstre og til højre for ordet *i*, og de 9 outputenheder angiver hver en af de 9 betydninger som jeg mener at ordet *i* kan have på dansk (deriblandt de 4 som er nævnt i eksemplet ovenfor).

Systemet af kategorier for ordene i konteksten til ordet *i* er følgende:

notation	Grovkategori som alle ord i ordbogen er blevet tilordnet
s	subst. der hører til kategorien sted/organisation/ikkehuman
k	subst. der hører til kategorien kvalitet
r	subst. der hører til kategorien relation/resultat
a	subst. der hører til kategorien aktivitet/begivenhed
c	subst. der hører til kategorien ciffer (tal)
i	præpositionen *i*
t	subst. der hører til kategorien tegn (semiotisk)
m	subst. der hører til kategorien mål (skala)
e	subst. der hører til kategorien emotion/kognition
d	subst. der hører til kategorien dato/tid
v	prædikat eller subst. der tager middelbart objekt med præpositionen *i*, eller et retningsadverbium
00	alt andet (inkl. nominer i genitiv)

Hvis man tager sætningen:

Han blev begravet -i- stilhed

kan den som input til det kunstige neurale netværk beskrives således:

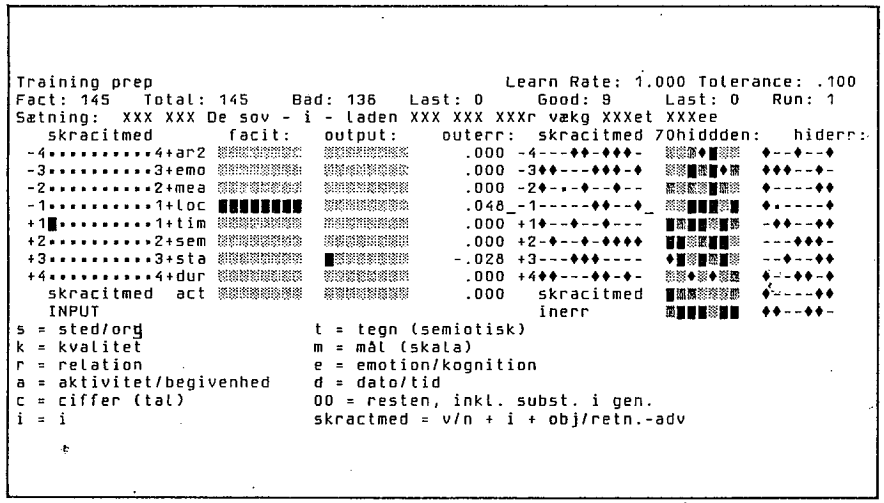
ord	kategori
-4:	00
-3:	00
-2:	00
-1:	00
+1:	r
+2:	00
+3:	00
+4:	00

Outputenhederne angiver de 9 forskellige betydningsvarianter af ordet *i;* Det drejer sig om (betydningerne af outputenhederne angives med engelske forkortelser i kapitæler, således at det er klart hvad der er sproglige eksempler og hvad der er sproglig betydning og hvad der er teknisk notation):

BETYDNINGSVARIANT AF ORDET *i*	NOTATION
1) objekt-markør, fx *bo i byen*: argument2:	AR2
2) 'med følelsen af', fx *i vrede*:	EMO
3) 'med et mål af', fx *i mange varianter*	MEA
4) 'på stedet', fx *i kælderen*	LOC
5) 'på tidspunktet', fx *i 1897*	TIM
6) 'betegnet ved', fx *i finansloven*	SEM
7) 'under en tilstand af', fx i hemmelighed	STA
8) 'under en varighed af', fx *i 4 dage*	DUR
9) 'under handlingen', fx *i kamp*	ACT

Skærmbilledet

Skærmbilledet med denne opstilling af det kunstige neurale netværk kommer så til at se således ud:



Linje 3 står den sætning som er input: *De sov - i - laden*; længst til venstre i det vertikale midterfelt er inputtet repræsenteret således som det er kodet, dvs. rækkerne der kaldes -4, -3 ...+3, +4 angiver hvilket ord i sætningen der er tale om, og i kolonnerne er det med *skracitmed* angivet hvilket kategori det pågældende ord hører til. Et ord af en given kategori på en given position, noteres så med en sort kasse. Notationen her angiver således at ord +1 (dvs. *laden*) hører til kategorien 's', dvs. STED/ORGANISATION. Kategorien verbum eller retningsadverbium noteres med udfyldning på pladserne *skractmed*.

Det næste område i det vertikale midterfelt beskriver det facit som jeg ønsker for denne sætning, nemlig at *i*'et i sætningen *de sov - i - laden* betyder 'stedet hvor', noteret LOC. På skærmbilledet er det angivet ved at alle 8 felter ud for LOC er udfyldt. Til venstre for facit er outputtet angivet. Man ser at det kunstige neurale netværk i dette tilfælde med 12% fyrer med outputenheden STA. Outputtet er altså helt forkert i forhold til det ønskede facit.

Til højre for outputkolonnen er der en kolonne tal med navnet *outerr*; det angiver i tal hvor stor fejl der er på hver enkelt output enhed sammenlignet med facittet.

Til højre herfor er det over navnet *inerr* angivet hvor stor fejlen er på hver af inputenhederne hvis outputtet havde været det rigtige. Jo sortere feltet er, des større fejl.

Næste kolonne angiver hvor meget input der har været til hver af de 70 skjulte enheder: jo sortere, desto større input. Endelig er det i sidste kolonne i midterfeltet angivet hvor store fejlene har været på de skjulte enheder i forhold til det ønskede facit.

Endelig er det i den nederste tredjedel angivet hvad bogstavkoderne i inputtet står for.

Træningen

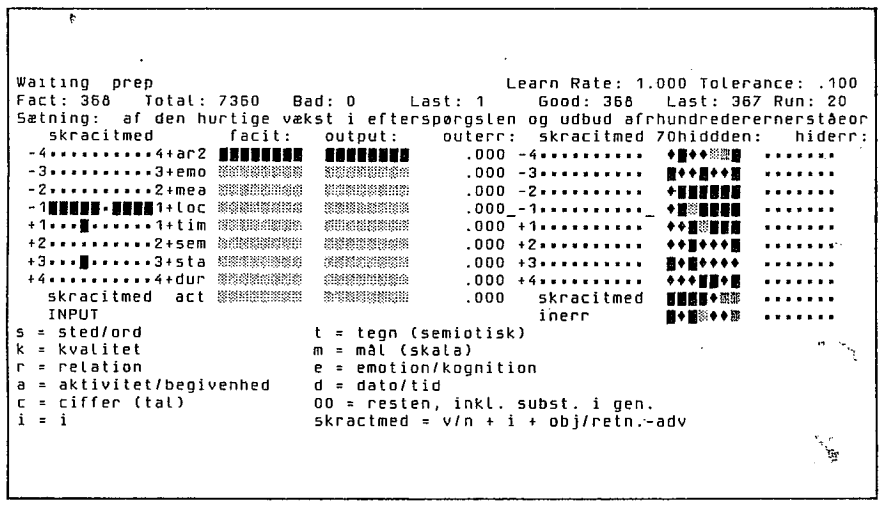
Jeg har nu trænet dette netværk med 368 sætninger der indeholder ordet *i* således som i eksemplerne med *han blev begravet i stilhed* og *De sov i laden*. Jeg har altså for hver af disse sætninger for det første kodet ind hvilken af de 12 kategorier hvert af de fire ord til venstre for *i*, og hvert af de 4 ord til højre for *i*, hører til, og for det andet givet facitlisten, dvs. hvilken variant af *i* der efter min mening er på spil. Her er et par af de autentiske eksempler jeg har trænet netværket med, og efter lighedstegnet angivelse af den rigtige betydningsvariant.

20. *sikre at hver deltager -i- samme projekt i hele* = AR2
21. *deltager i samme projekt -i- hele projektets løbetid til* = DUR
22. *dominere dette marked og -i- stigende omfang eksportere fra* = MEA
23. *nu er under overvejelse -i- nogle af de større medlemsstater* = LOC
24. *til et sådant nyt program -i- stor målestok er kommet* = MEA
25. *Esprit velkommen til mødet -i- juni 1992 og godkendte* = TIM
26. *XXX . Det sker -i- 1992 . XXX XXX* = TIM
27. *som anvendes i operationer -i- mange versioner og varianter* = MEA
28. *og afprøvning af VLSI/systemer -i- cilisium eller andre halvledere* = LOC
29. *beslutning end en afgørelse -i- raadet* = LOC
30. *fuldt kan støtte brugeren -i- kommunikationsprocessen,og som* = ACT
31. *; de vil resultere -i- nye produkter, processer* = AR2

32. *anvendelse foregår meget Langsommere -i- Europa end i Japan*
    = LOC

Det er klart at det kunstige neurale netværk i begyndelsen, hvor alle forbindelsesvægtene var stillet tilfældigt, gav helt forkerte output, som angivet i det første skærmbillede som viser sætning nr. 145 i første gennemløb.

Efter 20 gennemløb havde maskinen så "lært" alle eksemplerne. Dvs. vægtene var stillet således at outputtet i hver sætning blev således som jeg ønskede. Man kan se det på det næste skærmbillede:



Øverste linje *Waiting prep* angiver at det stadig er den samme fil, nemlig *prep*, der kører, men ordet *waiting* angiver at træningen nu er færdig. Man ser at det er sætning nr. 368, *af den hurtige vækst - i - efterspørgslen og udbud af* ... I 2. linje kan man desuden fra højre til venstre læse at det er gennemløb nr. 20, at der ved sidste gennemløb var 367 rigtige, at der i dette 20. gennemløb har været 368 rigtige, dvs. alle sammen, at der i sidste gennemløb var én forkert, men at der i dette har været 0 forkerte, at der i alt har været trænet på 7360 træningssætninger.

Ellers ser man i det vertikale midterfelt i venstre kolonne: det kodede input, dernæst facit: AR2, output: 100% AR2, ingen outputfejl, ingen inputfejl, fyringsmønstret i de skjulte enheder, og ingen fejl i de skjulte mønstre.

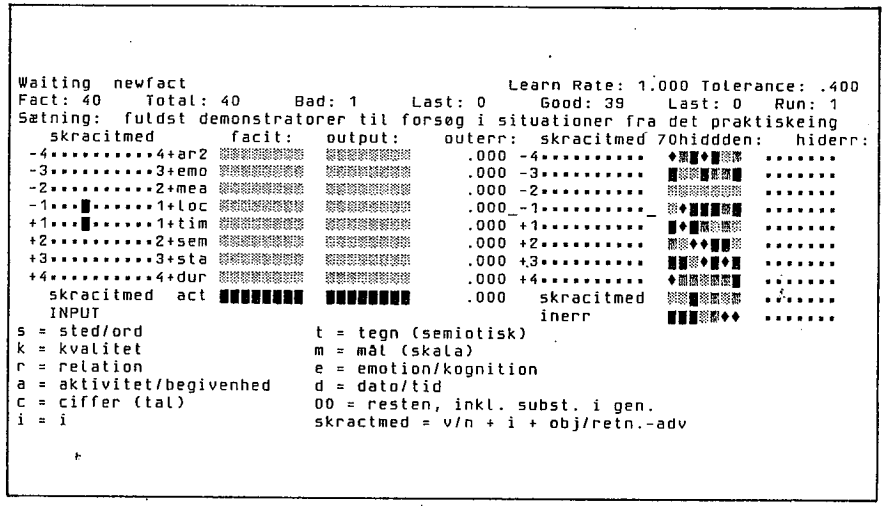
Det neurale netværk har nu fået justeret alle sine vægte således at det kan give det rigtige output på alle de 368 sætninger som træningssettet består af.

Reglen

Det interessante er nu at maskinen ved justeringen af vægtene faktisk må have foretaget en generalisering, dvs. lavet en repræsentation af en regel, for nu kan netværket også lave korrekte bestemmelser af betydningen af ordet *i* i sætninger som ikke indgik i træningseksemplerne, altså fx:

Sektorer, som i mindre grad umiddelbart påvirkes
af silicium med dimensioner i området 1 my,
den omfatter også forskning i optoelektronik, optiske digitale
projekter indkaldes og udveksles i overensstemmelse med de relevante
som oversendes til Rådet i overensstemmelse med de relevante
databaser som kan anvendes i praksis
forskellige problemer, men i princippet finder kommissionen,
- og må ind i privathjem, kontorer og
undersøgelser som blev påbegyndt i slutningen af sidste år

Netværket kunne bestemme betydningsvarianten af *i* fuldt korrekt i 39 af 40 autentiske eksempler som ikke indgik i træningseksemplerne. Man må således sige at maskinen kan løse opgaven fuldt tilfredsstillende. Denne præstation dokumenteres af følgende skærmbillede:



Af den første linje ser man at det er en ny fil, *newfact*, den er færdig med. Den har lavet ét gennemløb og er nået til sætning nr. 40: *...fuldstændige demonstratorer til forsøg i situationer fra det praktiske* ... Outputtet har i denne sætning været helt korrekt, og man kan se at det har det været i 39 af de 40 sætninger.

Man ser at det kunstige neurale netværk bruger en del tid på træning; i dette eksempel er der jo 5600 vægte der skal justeres ved hvert træningseksempel mellem inputlag og skjult lag, og 630 mellem skjult lag og outputlag.

Men når dernæst alle vægtene er justeret, løser netværket opgaven at entydiggøre ordet *i*, hurtigt og elegant. Det tager jo nemlig kun en brøkdel af et sekund for det trænede netværk at afgøre hvilken af de 9 betydninger der er på spil i et givet eksempel. De 40 eksempler har den entydiggjort på i alt 3 sekunder.

Man må nu forestille sig en maskine der består af forskellige moduler: Først er der et kunstigt neuralt netværk med en ordbog der har alle ordene grovsorteret, dvs. hvori det for hvert ord er angivet hvilken af de 12 kategorier det hører til. Det neurale netværk afgør så i hvert tilfælde hvor det støder på ordet *i* hvilken variant af i der er på spil i præcis denne kontekst. I vores eksempel vil det give følgende resultat:

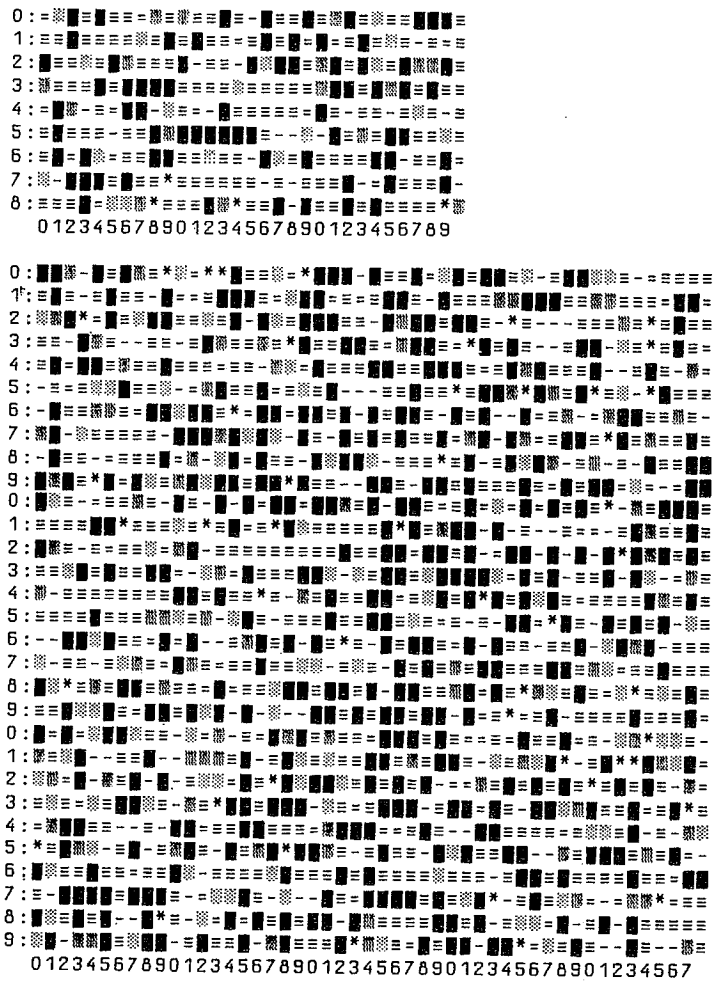
*Lenin boede i*_TIM *1897 i*_STA *hemmelighed i*_AR2 *Kbh i*_DUR *4 dage.*

Denne forarbejdede inputsætning køres dernæst igennem parseren med grammatik og ordbog der har 9 forskellige ord der hedder *i*, præcis som i den serielle maskine. Outputtet af denne analyse bliver de 8 syntaktiske mønstre (som dog pga. de modificerede input reduceres til 4), mens ingen af de øvrige 6561 analysetræer vil blive lavet, fordi det i inputtet er angivet hvilken af de 9 *i*'er der er på spil i hvert enkelt tilfælde.

Hvis dette system skal køre, skal der være trænede kunstige neurale netværker for hver af de præpositioner der kan have ca. 10 betydninger; der skal altså være 10-15 forskellige netværker. Derimod behøver der næppe at være trænede netværker til at entydiggøre flertydige substantiver som næppe har mere end 5 betydningsvarianter, eller til flertydige verber som ofte kun har et par betydningsvarianter.

Reglens repræsentation

Når netværket kan analysere endnu ikke prøvede sætninger rigtigt - næsten - hver gang, må det være fordi maskinen på en eller anden måde har en repræsentation af den regel som blev brugt i den serielle maskine. Reglen må ligge i det mønster af vægte der er på forbindelserne mellem enhederne i de tre lag. Her følger et skærmbillede der viser hvilken vægt der er på hver af de 630 vægte der er mellem det skjulte lag og outputlaget (øverst) og hver af de 5600 vægte der er mellem inputlaget og det skjulte lag (nederst); jo sortere feltet er, desto stærkere er forbindelsen.



Det er jo svært at se; men der må fx være en regel der siger at hvis ord +4 hører til kategorien relation/resultat, så er betydningen af ordet *i* i_sta, dog kun hvis alle ordene +1 til +3 alle hører til kategorien resten (00). Reglen er der, men det er svært at se hvor den er.

Og mest interessant: hvis jeg træner netværket endnu engang fra bunden, med de samme 368 træningssætninger, vil det have lært reglen på præcis 20 gennemløb - som første gang, og den vil give det rigtige output på 39 ud af de 40 samme kontrolsætninger som første gang. Dvs. man kan med god ret sige at det kunstige neurale netværk har lært præcis den samme regel. Men kigger man efter på forbindelsesvægtene, bliver billedet helt anderledes.

Reglen ligger altså repræsenteret i det trænede neurale netværk som et meget abstrakt mønster af relationer mellem vægtene på alle forbindelserne.

Der ligger således ikke nogen del af reglen på nogen af de forskellige forbindelsesvægte. Hele reglen ligge på alle forbindelsesvægtene på en gang, nemlig i mønstret af relationer mellem de enkelte forbindelsers vægte.

Parallelle og serielle maskiner

Denne repræsentationsform betyder, sammenlignet med den serielle maskine flere forskellige ting: det trænede netværk er meget hurtigere end den serielle selektionsmekanisme; det trænede netværk er meget mere robust end det serielle program, dvs. fejl i inputtet giver kun forholdsmæssige fejl i outputtet; og resultatet af det kunstige neurale netværk er altid graderet således at man ikke blot finder mulige output og udelukker umulige output, men også således at maskinen finder det bedste blandt de mulige output.

Den serielle kunstig intelligens-maskine ville til sammenligning bruge timer på at opstille alle de træstrukturer som ville være mulige. Med det tilsvarende antal betydningsvarianter ville maskinen opstille: 9 x 9 x 9 x 9 x 8 = 52.448 analysetræer for eksemplet: *Lenin boede i 1897 i hemmelighed i København i 4 dage.* Dernæst skal den så sammenligne hvor tæt den semantiske relation er i hver af de forbindelser der er mellem hver af de konkurrerende præpositions-

varianter og det tilhørende nomen (styrelsen), for til sidst at vælge den med den bedste semantiske relation.

Den serielle maskine er meget 'skør' sammenlignet med det parallelle program. Hvis der i inputtet er en stavefejl, eller hvis der i inputsætningen er en metafor (som kan beskrives som et brud på selektionsreglerne), vil maskinen og herunder selektionsmekanismen) slet ikke give noget output. Det neurale netværk vil derimod altid give et eller andet output, og det vil altid være det mindst ringe. Det kan godt være at det ikke er det rigtige - men det er jo også i metaforer diskutabelt hvad der er det 'rigtige'.

Når den serielle maskine kører på formulerede regler som manipulerer med symboler, så kan den også kun anvende regelmæssigheder der går på symbolerne. Men der findes faktisk subsymbolske regelmæssigheder, fx statistiske tendenser - også i entydiggørelsen af betydningsvarianter af præpositioner. Fx er det en overvældende statistisk regel at hvis der efter et *i* kommer et talord, så er det betydningen DUR, fx *i 4 dage*; men det er ikke nogen regel som det serielle program kan bruge, og det er en statistisk regelmæssighed som det er så indviklet at beskrive i forhold til symbol-manipulationsreglerne at det slet ikke kan betale sig at forsøge at bygge en statistisk tendens ind i regelsystemet.

Denne statistiske regelmæssighed er automatisk en del af det kunstige neurale netværk, som generelt kan beskrives som en statistisk beskrivelse af regelmæssigheder.

Generelt kan man altså sige at der findes opgaver hvortil parallelle programmer (kunstige neurale netværker) egner sig meget bedre end serielle programmer (klassiske Kunstig Intelligens-programmer), nemlig de opgaver som går ud på at kategorisere et input. Til disse opgaver er de parallelle programmer hurtige, robuste og effektive, mens de serielle er langsomme, skøre og ueffektive.

Men det skal også siges at kunstige neurale netværker ikke kan bruges til alle opgaver. Man kan fx næppe lave en fungerende parser alene med parallelle programmer.

Rumelhart og McClelland giver ellers i deres bog et par eksempler på hvorledes de forestiller sig af parsningsproblemer løses med parallelle programmer, men de er mildest talt urealistiske og naive. De synes nemlig fuldstændig at glemme at der til en realistisk parser skal bruges en ordbog med 100 000 ord, og at grammatikken skal

være indrettet således at den kan analysere sætninger med principielt uendeligt mange ord efter hinanden. Det er således ikke et bevis for noget som helst - end ikke et eksempel på noget som helst, at de med en ordbog på 20 ord kan analysere en 3-ledssætning som *The boy hit the girl*. For at kunne parse sætninger fra naturlige sprog skal man have et system af regler der manipulerer symboler, og som tillader rekursive regler, dvs. regler der tager deres eget output som input. Alene disse regler kan nemlig gøre rede for sætningernes principielle uendelighed.

Får man så noget for intet?

Det er en almindelig opfattelse at man med brug af kunstige neurale netværker så at sige får noget for intet. Man har et problem man ikke kan løse. Man ved hvad input er, og man ved hvad man vil have som output, men man ved ikke hvordan man kommer fra input til output, dvs. man kan ikke lave nogen regler der kan simulere processen således som mennesker udfører den. Så laver man et neuralt netværk, træner det, og vupti, så har man en løsning, nemlig et neuralt netværk der kan simulere processen. Man ved ganske vidst ikke hvordan det neurale netværk fungerer, men det fungerer, og på den måde har man fået noget for intet.

Jeg har prøvet i det foregående at vise at dette ikke er en sandfærdig historie. Selv om man ud fra et overfladisk synspunkt kan hævde at jeg godt kunne have lavet mit parallelle præpositionsprogram også uden at jeg selv havde lavet det serielle selektionsprogram, vil en nøjere overvejelse vise at det ikke er tilfældet. Jeg har jo nemlig meget omhyggeligt udvalgt hvilke træk ved inputtet (hele sætningens udtryksside) som jeg ville opfatte som så relevante at de kunne være af betydning for kategoriseringen af det. Og det kunne jeg ikke have gjort hvis jeg ikke - i det mindste i store træk - havde kendt reglen på forhånd.

Man vil se at hvis jeg ikke havde vidst at det var de tolv kategorier sted/kvalitet/relation... der var de relevante, ville jeg have være henvist til at indkode træningssætninger ved deres bogstaver. Dvs. jeg kunne ikke have klaret mig med 8 x 10 procesenheder i inputlaget, men havde måttet bruge ca. 50 bogstavpladser (25 pladser før *i* og 25

pladser efter) med mulighed for 29 bogstaver på hver plads, altså 1450 enheder i inputlaget. Og tænk så på hvor mange træningssætninger jeg skulle have haft, for at alle tilfældige variationer skulle have været neutraliseret således at netværket kun havde lært reglen om hvilke betydninger af præpositionen der passer til hvilke kontekster. Det havde ganske simpelt været uendeligt.

Med mit neurale netværk har jeg altså ikke fået noget for intet. Jeg har fået et hurtigt, robust og effektivt program til løsning af en bestemt delopgave i maskinoversættelsen, nemlig entydiggørelsen af ordet *i* på grundlag af dets kontekst. Og jeg har kun fået det fordi jeg i store træk vidste hvordan problemet kunne løses således at jeg har kunne sortere inputtet i lige præcis de relevante enheder. Men jeg har faktisk fået noget der er meget bedre end det serielle program som jeg med den samme viden om problemets løsning kunne konstruere.

Kunstige neurale netværker er således ikke epokegørende nye typer af maskiner, men en smart måde at programmere løsninger af bestemte opgaver på.

Subsymbolske processer

Det skal til sidst bemærkes at der ikke er nogen filosofiske implikationer af de her fremførte fordele ved parallelle processer i forhold til serielle. De neurale netværker der er omtalt her er nemlig kørt på ganske almindelige PC'ere, som jo kører serielt. Jeg og andre der laver parallelle programmer, simulerer altså blot parallelle programmer på en seriel maskine. Det har intet med grundlæggende analoge og digitale repræsentationer at gøre. Det hele er i bunden digitalt og serielt.

Man kan beskrive det således at de parallelle programmer og de serielle programmer der løser den samme opgave, nemlig at entydiggøre præpositionen *i* i dens kontekst, er to forskellige måder at programmere **den samme** regel på. Det kræver selvfølgelig noget argumentation, men jeg hævder at det er den samme regel jeg har programmeret ind i de serielle og det parallelle program.

Lingvistisk kan man beskrive reglen således at præpositionen *i* som leksikalsk størrelse har mulighed for i en kontekst at komme til

at betyde 9 forskellige ting. Betydningen i en konkret position i en konkret sætning (dvs. en kontekst) afhænger af om der før præpositionen kommer et prædikat (verbum, adverbium eller adjektiv) der styrer præpositionen *i* (fx *bo_i, ud_i*), og hvilken semantisk kategori det efterfølgende nominal (der er præpositionens såkaldte *styrelse*) hører til.

Denne lingvistiske regel kan nu programmeres således at maskinen analyserer hvilke prædikater der styrer præpositionerne og hvilke nominaler der er styrelse for præpositionerne (det er uheldigt at det nominal der kommer efter præpositionen hedder *styrelsen* på dansk, men det gør det). Først når denne analyse er foretaget og analysens resultat er repræsenteret i maskinen i form af symboler for de enkelte dele og forbindelser i præpositionens kontekst, kan den tjekke i hvilke af de producerede konkurrerende repræsentationer der er match mellem præpositionens selektionsrestriktioner og den semantiske kategorier i de relevante grammatiske led i konteksten. Det er det serielle program.

Den samme lingvistiske regel kan også programmeres således at maskinen ud fra en meget grovmasket klassifikation af konteksten som helhed afgør hvilken af de mulige konkurrerende betydninger af præpositionen der har det bedste match med den samlede kontekst. I det parallelle program karakteriseres konteksten altså som en helhed uden brug af symbolske repræsentationer af de enkelte dele og strukturer.

I det serielle program foregår entydiggørelsen af ordet *i* som en beregning af et matematisk problem ved manipulering af symboler. I det parallelle program foregår entydiggørelsen af ordet *i* som genkendelsen af en helhed ved en gestaltning. Det er jo denne måde mennesker opfatter betydningen af et ord på. Når vi hører de 4 lyd h-e-s-t efter hinanden, foretager vi jo ikke en beregning af hvad indholdet af dette tegn kan være ved at manipulere med de symboler der indgår. Vi genkender helheden som en gestalt. På samme måde kan man sige at maskinen genkender konteksten til ordet *i*, dvs. hele den sætning ordet indgår i, som en gestalt.

En af de væsentlige fordele ved det menneskelige sprog er den såkaldte dobbelte artikulation. Vi har tegn, dvs. symboler som vi kan manipulere med i syntaksen, det er et fast genkendeligt udtryk, fx *hest*, som har en stabil relation til et indhold: 'hest'. Men vi har også

lyd og bogstaver, som ikke er tegn eller symboler, og som vi ikke kan manipulere med; det er subsymbolske figurer som i sig selv kun har adskillende funktion. Og man kan ikke nå fra ordets bogstaver, ved manipulation med dem til betydningen af ordet; det er en helhedsgenkendelse, og man plejer at kalde det en arbitrær relation mellem udtryk og indhold. Det er jo ikke en genkendelse af en række af enkelte lyd, eller en række af sorte streger på et papir, men af et abstrakt mønster mellem disse udtryksfigurer, således at genkendelsen er den samme af de 4 lyd *h-e-s-t* og de 4 bogstaver *h e s t*.

Det menneskelige sprog effektivitet beror på denne dobbelte artikulation. Vi har ved et meget begrænset inventar af subsymbolske figurer (nemlig 29 bogstaver eller ca. 40 lyd) midler til at udtrykke 100 000 forskellige tegn eller symboler som vi dernæst kan manipulere med. Det gør det muligt med vores mund at artikulere alle de 100 000 begreber forskelligt og dog genkendeligt. Det gør det muligt at lave skrivemaskiner. Tænk på hvordan en kinesisk skrivemaskine ser ud, det er en hel lagerhal med alle de rammer der skal til for at man har en type til hver af de titusindvis af forskellige tegn der findes. Og tænk på hvordan vores mund skulle trænes hvis vi skulle kunne udtrykke ikke ca. 40 forskellige lyd med den, men 100 000 lyd.

Det er altså denne opfindelse af den dobbelte artikulation der overhovedet gør det muligt for os mennesker at transmittere 100 000 forskellige ordbetydninger (og milliarder forskellige sætningsmeddelelser) fra den ene bevidsthed til den anden gennem et fysisk medium (luft molekylernes svingninger i lyden, eller bogstaverne i skriften). Uden den ville vore kommunikation være begrænset til de 20-30 forskellige meddelelser som mange dyrearter kan kommunikere til hinanden gennem et fysisk medium (fx fiskenes farver og bevægelser).

Serielle programmer er gode til at eftergøre det arbejde som vi i sproget gør ved hjælp af syntaksen, nemlig at manipulere med symboler - for det serielle program er i sig selv en manipulation med symboler. Parallelle programmer er gode til at eftergøre det vi mennesker gør med sprogets subsymbolske niveau, nemlig at transmittere meningerne med det enkelte symbol - for det parallelle program er i sig selv gestaltningsprocesser på grundlag af subsymbolske figurer (forbindelsesvægtene).

Menneskets sproglige forarbejdninger i kommunikationsprocessen er selvfølgelig både gestaltning af subsymbolske figurer og manipulation med symboler. På samme måde vil den bedste maskinsimulering af menneskets sproglige processer (som det forsøges i fx maskinoversættelse) være en kombination af serielle og parallelle programmer.

Man kan så i den filosofiske diskussion af hvad tænkning og bevidsthed er tilføje at det ser ud til at neurale netværker simulerer de processer i hjernen som altid er ubevidste, men hurtige, robuste og effektive kategoriseringer som har uendeligt mange forskellige input, men et begrænset antal output, mens serielle programmer simulerer de processer i hjernen som ikke behøver at være ubevidste, men som kan gøres bevidste og genstand for bevidst viljesstyring, og som indeholder en kombinatorik der kan producerer uendeligt mange forskellige output på grundlag af et begrænset antal input.

Man kunne endelig forestille sig at disse forskellige processer - der har vidt forskellige input-output-relationer, havde forskellige fysiske realisationer i den menneskelige hjerne. Med andre ord: de virkelige neuroner i hjernen fungerer antageligt forskelligt i de to typer af processer.

Litteratur

BrainMaker. Users Guide and Reference Manual. 3rd Edition, January 1989. Sierra Madre, CA.

Rumelhart, David E., James L. McClelland and the PDP research Group: *Parallel distributed processing. Explorations in the Microstructure of Cognition. Vol 1-3*, 1986 MIT Press, Cambridge Mass.